

Opinnäytetyö Turun AMK
Tietojenkäsittelyn koulutusohjelma
Sähköisen liiketoiminnan järjestelmät
2015

Aleksi Myllymäki

KODUN ALKEIDEN PILOTOINTI SALON SEUDULLA



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma | Sähköisen liiketoiminnan järjestelmät

2015 | 15 Sivua + 40 liitesivua

Ohjaaja Päivi Killström

Aleksi Myllymäki

KODUN ALKEIDEN PILOTOINTI SALON SEUDULLA

Koska ohjelmoinnin merkitys lisääntyy kouluissa tulevaisuudessa, tarvitaan käyttöön erilaisia ohjelmia ja oppimismenetelmiä. Tässä työssä käsitellään Kodu-nimistä ohjelmaa, jota on käytetty ohjelmoinnin logiikan opettamisessa Salon seudulla. Kahden vuoden aikana on selvitetty alakoulun oppilaiden osaamista ja kiinnostusta ohjelmointia kohtaan. Projektissa on käsitelty Kodun alkeita ja oppimisen kannalta oleellisia ohjelmoinnin logiikkaa opettavia toimintoja.

Tässä työssä on suunniteltu itselaadittu materiaali Kodu-ohjelman käyttöön ja alkeisopetukseen. Työssä käsitellään myös tulevaisuuden suunnitelmat ohjelman hyödyntämiseen ja miten ohjelmaa voitaisiin kehittää jatkossa opetusta varten.

Itselaaditun oppaan testaus toteutettiin hyödyntäen neljää eritasoista ohjaajaa opetuksessa. Haastattelut toteutettiin avoimella keskustelulla. Haastattelun tulokset kertovat ohjeen helppokäyttöisyydestä. Ohjaajien kehitysideat koskevat mm. käännöstyötä ja Kodun hyödyntämistä aineenopetuksessa, kuten esimerkiksi matematiikan.

Tulevaisuudessa pitäisi kehittää ohjetta haastavampien pelien suuntaan. Ohjelman algoritmia ja ominaisuuksia tutkimalla saataisiin materiaalia mahdollisesti myös yläkoulun käyttöön. Ohjelman ja koulutuksen kehityksen kannalta on tärkeää saada potentiaaliset henkilöt jatkamaan Kodun tai muun ohjelmoinnin parissa..

ASIASANAT:

Kodu, ohjelmointi, varhaiskasvatus, koulutus, opetus

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Business Information Technology | e-Business Systems

2015 | 15 Pages + 40 attachment pages

Instructor Päivi Killström

Aleksi Myllymäki

KODU PROJECT PILOTING IN SALO REGION

The meaning of programming at schools is becoming more important in the future and there are different kind of programs and teaching methods available nowadays. This thesis handles the program called Kodu that has been used in teaching programming logic in Salo region. The interest and know-how of elementary school students when using Kodu in programming has been researched for two years with the aim of finding out the basics and important points they need for learning the program.

During this thesis work self-made material or manual is developed for learning and teaching of Kodu. The thesis also considers future plans for using the material and its further development for future teaching.

The testing of self-made manual was carried out by exploiting four instructors with different levels of know-how and teaching skills. The interview was an open conversation with the instructors. The results of the interview indicate that the manual or material is easy to read and understand and the results also offer improvement ideas for translating and using Kodu in subject-related studies.

The manual contents should be developed towards more advanced games by studying the program's algorithm and properties. That way it would be possible to get material also for the use of secondary school. This is important for developing the program and the education and also for getting potential persons to continue the use of Kodu or other programming.

KEYWORDS:

Kodu, programming, early childhood education, teaching, education

SISÄLTÖ

1 JOHDANTO	5
2 YLEISIVISTÄVÄÄ OHJELMOINTIA KODULLA	6
3 ALKEISOPAS JA SEN TESTAUS	11
3.1 Oppaan testaaminen ohjaajilla ja testauksen tulokset	11
3.2 Kolme kehitysideaa	12
4 YHTEENVETO	14
LÄHTEET	15

LIITTEET

Liite 1. Opas Kodun alkeisiin.

1 JOHDANTO

Nykypäivänä ohjelmoinnin ja logiikan ymmärtäminen on tärkeää jo yhä nuoremmille. Vähän väliä voidaan lukea lehdistä ja kuulla televisiosta ”koodareista”, jotka ovat apuohjelmilla oppineet ohjelmoimaan tai ymmärtämään logiikkaa. Ohjelmoinnin tärkeys korostuu nykyään jo yhä nuoremmilla ja valtio yrittää myös toimillaan vauhdittaa asian etenemistä hankkimalla esimerkiksi tabletteja ja muuta tietotekniikkaa kouluihin. Myös ohjelmoinnin opetusta yritetään lisätä kouluissa esimerkiksi vapaaehtoisten opettajien toimesta ja erilaisin tempauksin. Ohjelmoinnin ymmärtämisen lisääminen on myös omiaan tulevan opetus-suunnitelma muutoksen johdosta 2016.

Itse olen toiminut Kodu-ohjelmoinnin ohjaajana Salon seudulla muutaman vuoden opiskeluiden ohessa. Olen huomannut lasten ja nuorten potentiaalin ja innostuksen ohjelmointia kohtaan. Tähän tarvitaan sopivat työvälineet, jossa Kodu onnistuu hyvin sen helppokäyttöisyyden ja visuaalisen käyttöliittymän takia.

Salossa Kodua lähdettiin kokeilemaan seudun kouluissa vuonna 2013 alussa. Kevääseen 2014 mennessä kouluja oli opetuksen piirissä jo 12 suuren suosion ansiosta, joissa opetus tapahtui iltapäiväkerhotoimintana. Koulut saatiin mukaan esittelytilaisuuksilla joita järjestettiin koulujen rehtoreille ja ATK-opettajille. Ilta-päiväkerhoja vedettiin aina kahden ammattikorkeakouluopiskelijan johdolla. Tulevaisuutta varten päätin kehittää alkeisoppaan uusille tuleville ohjaajille, joka olisi helposti ymmärrettävissä taitotasosta riippumatta, on kyseessä sitten ohjelmointialan opiskelijat tai peruskoulun opettajat.

Työssä on myös selvitetty eritasoisten ohjaajien kouluissa hyödyntämään valmista ohje- tai opetusmateriaalia, sekä heidän kommenttinsa siitä. Millaisia kehitysideoita heillä on ohjelman menestymiseksi tulevaisuuden opetuksessa, sekä käydään läpi kolme tulevaisuuden kehitysideaa perusteellisemmin.

2 YLEISIVISTÄVÄÄ OHJELMOINTIA KODULLA

Valtio on muuttamassa opetussuunnitelmia valtakunnallisesti ja ohjelmoinnin on tarkoitus tulla peruskouluihin jo 2016. Normaalien opettajien olisi tarkoitus alkaa opettamaan ohjelmointia aina ensimmäisestä luokasta yhdeksänteen asti.

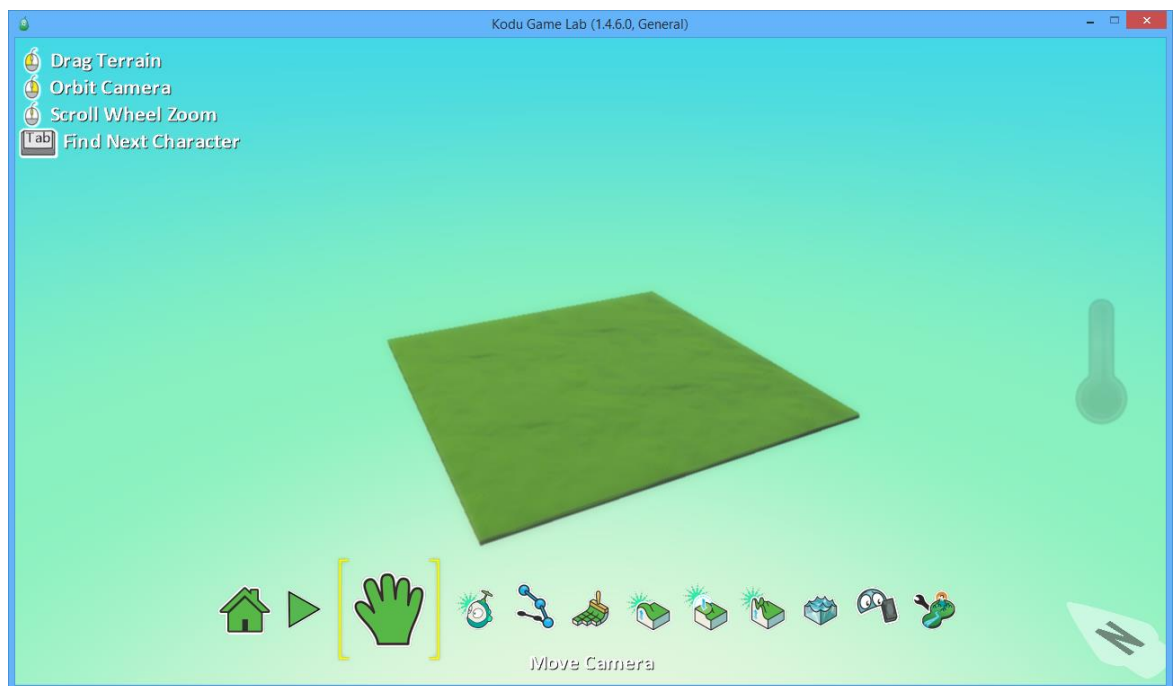
Kouluissa tulisi opettaa ohjelmointia yhtä yleissivistävästi kuten englantia tai maantietoa. Tulevaisuudessa kuitenkin yhä useampi puhelin tai tietokoneohjelma tehdään juurikin ohjelmoimalla ja sen ymmärtäminen korostuu yhä nuorempana. Kaikista ei voi tulla ohjelmoijia, mutta ei myöskään geologeja tai matemaatikkoja. Tarkoitus on luoda mahdollisuus kaikille oppia ja ymmärtää ohjelmoinnin mahdollisuudet eri aloilla ja asioissa, sekä merkitys tulevaisuuden kilpailukyvyn kannalta. (Mykänen & Liukas, 2014, 8–9.)

Salon seudulla ohjelmointi on lähtenyt sekä Microsoftin että paikallisen yrityshautomon ajatuksesta, miten ohjelmointia voitaisiin lähteä kokeilemaan Salossa ja kuinka sitä voitaisiin hyödyntää aluksi Salon alakouluissa. Tällöin ajatukseksi muodostui Kodu-ohjelma, jota on käytetty ohjelmoinnin perusteiden opetukseen esimerkiksi Iso-Britanniansa. Siellä on oma yhteisönsä Kodun opetuksessa ja sen piiriin on kuulunut jopa 75 000 lasta ja nuorta. (Evans 2013.)

Ohjelmoinnin opetukseen löytyy sekä maksullisia että ilmaisia ohjelmia. Tässä työssä keskityttiin Kodu-ohjelmaan, joka soveltuu loogisten taitojen kehittämiseen. Oppimisen alkuvaiheessa se motivoi oppilaita leikillisen ilmeensä ansiosta. Perusohjelmointikieleen verrattuna Kodu on helpommin lähestyttävä ja soveltuu myös nuoremmille henkilöille, koska ohjelmointi ei tapahdu perinteisesti rivikoodaamisella vaan visuaalisilla komennoilla.

Kodu ohjelmana

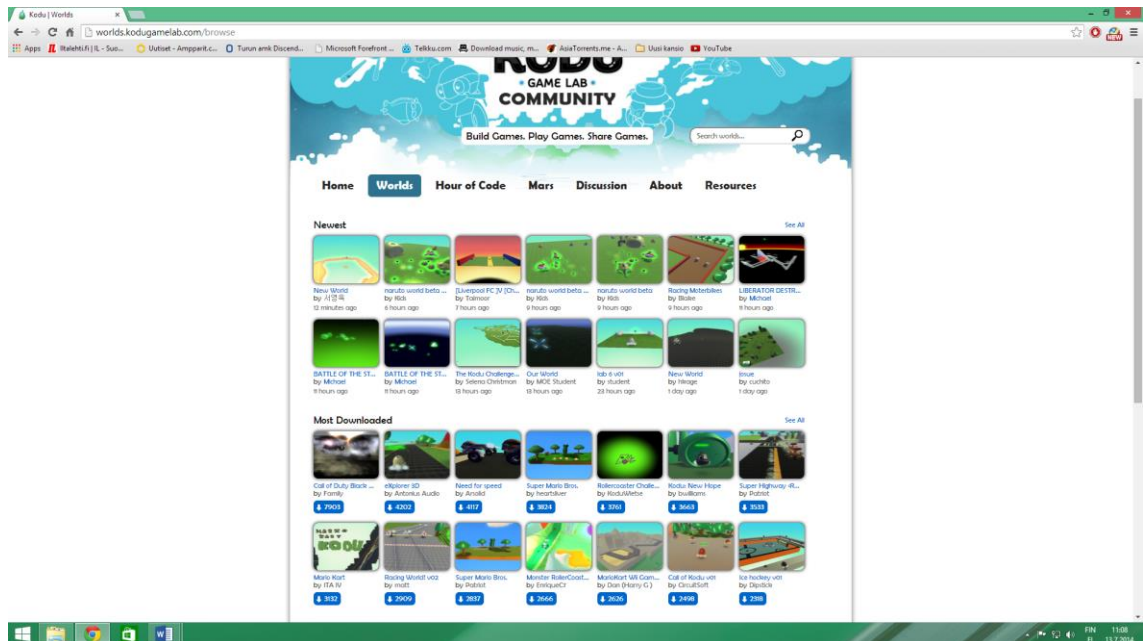
Kodu on Microsoft FUSE Labsin kehittämä visuaalinen ohjelmointiympäristö. Ohjelma julkaistiin vuonna 2009 ja on vuosien saatossa levinnyt maailmalla ja saanut enemmän sisältöä päivitysten avulla. Kodu on visuaalinen ohjelmointikieli, jonka avulla kuka tahansa voi helposti luoda omia 3D-pelejä (kuva 1). Henkilöllä ei tarvitse olla osaamista itse ohjelmoinnista ennen käyttöä. Kodun idea ei perustu vain ohjelmoinnin opettamiseen, vaan myös loogisen ajattelun, ongelman ratkonnan, tarinan kerronnan ja visuaalisten taitojen opetteluun. (Kelly 2013, 4.)



Kuva 1. Kodun visuaalinen käyttöliittymä.

Kodu on ohjelmana kuitenkin valitettavan rajallinen verrattaessa yleisimpiin ohjelmointikieliin, mikä voi muodostua ongelmaksi myöhemmin. Suurin ja yleisin rajoittaja Kodussa on objektien luonti. Maailmassa voi aina olla vain tietty määrä objekteja, tai vastaavasti maailma voi olla vain tietyn kokoinen. Tämä ongelma yleisesti ratkotaan rakentamalla oma maailma osittain moniin tasoihin. Määritellyllä muuttujalla tapahtuma vie pelaajan seuraavaan maailmaan tai tasoon.

Vanhemmilla ja vähätehoisilla tietokoneilla voi olla ongelmia ison maailman suorittamisessa.



Kuva 2. Erilaisten pelien lataaminen kodugamelab.com-sivustolla.

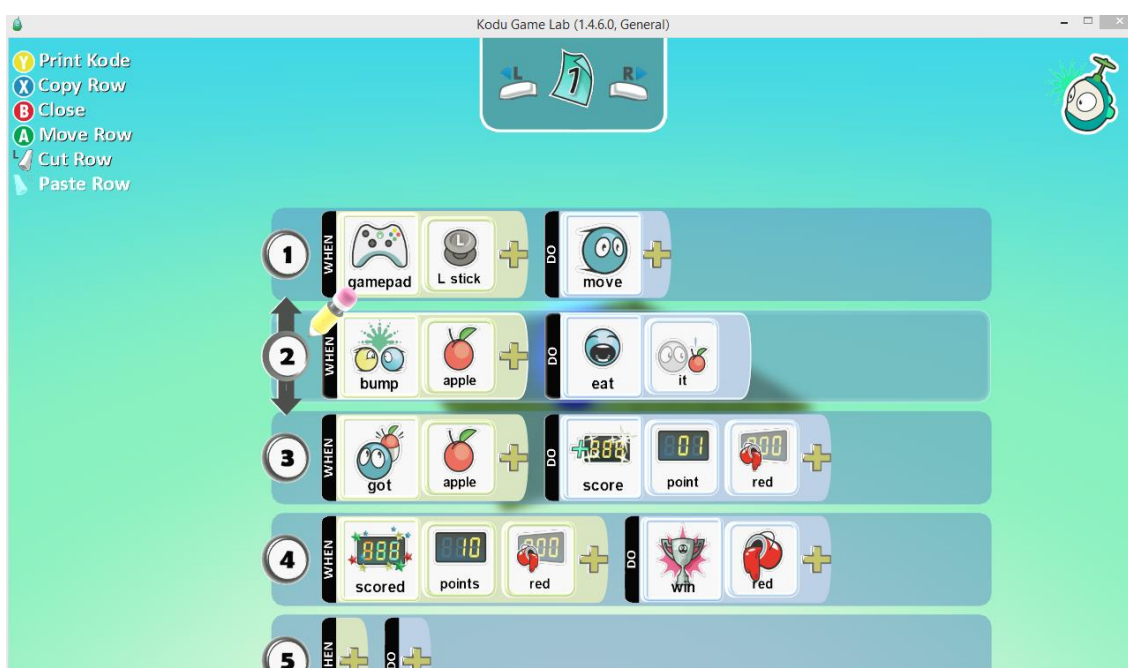
Kodulla tehtyjä pelejä pystytään jakamaan verkossa erilaisissa yhteisöissä ja foorumeilla, kuten esimerkiksi Microsoftin ja Kodu Gamelabin foorumeilla tai kotisivulla (kuva 2). Kyseisten foorumeiden käyttökieli on tällä hetkellä vain englanti. Kodu perustuu paljolti yhteisöllisyyteen ja muiden tukeen ongelmatilanteissa kyseisten verkkosivujen kautta. Avustusta on täten mahdollista saada muilta tekijöiltä tai käyttäjiltä. Foorumeilta löytyy paljon valmiita opetuspelejä monimutkaisempien pelien suunnitellun harjoittelemiseen. Yhteisöllisyys ja avoin ohjelma mahdollistaa näin muiden tekijöiden pelien hyödyntämisen omassa pelissä.

Ohjelmointi Kodulla

Kodulla ohjelmointi on yksinkertaista visuaalisen käyttöliittymänsä ansiosta kuten aikaisemmin mainittiin. Käyttö ei siis vaadi minkäänlaista osaamista ohjelmoinnista tai sovelluskehityksestä ennestään. Kaikki komennot ja suunnittelu

tapahtuu helpolla visuaalisella käyttöliittymällä jota lapsi ja aikuinen ymmärtää. Itse tekeminen tapahtuu joko Xbox-ohjaimella tai hiiri/näppäimistö - yhdistelmällä. Näytöltä voidaan aina tarkistaa, mitä mikäkin painike tai näppäinyhdistelmä tekee. Uusin muutos ohjelmassa on mahdollisuus käyttää Kodua kosketusnäytöllä. Windows 8:n jälkeen ohjelmaan on tullut päivitys kosketusnäytöille, joka on omiaan kosketuslaiteille kuten Windows RT/Pro -tableteille. (Kodu 2015.)

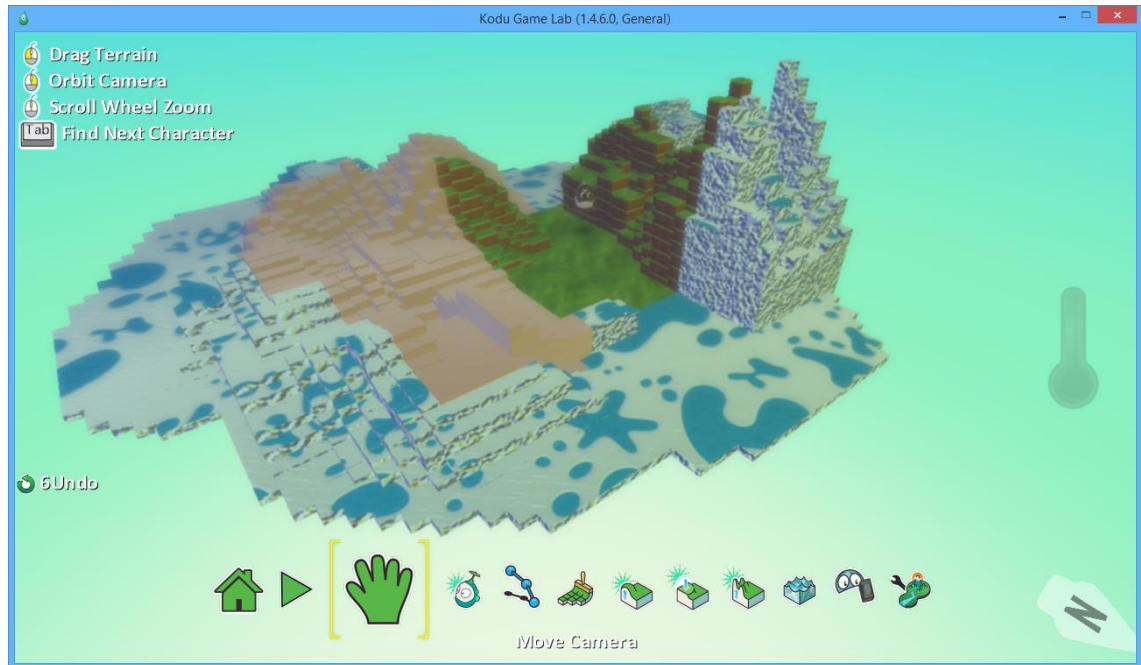
Oheisessa kuvassa (kuva 3) näytetään mitä visuaalisella käyttöliittymällä käytännössä tarkoitetaan. Ohjelmointi tapahtuu ehtolausekkeiden avulla ja jokaisella lausekkeella on aina jonkinlainen syy ja seuraussuhde.



Kuva 3. Objektien ohjelmointi.

Kuvassa hahmo on ohjelmoitu liikkumaan vasenta sauvaa ohjaimesta liikutettaessa johonkin suuntaan. Lisäksi hahmoon on ohjelmoitu, että kun se törmää omena, se syö sen ja saa siitä yhden punaisen pisteen "got"-komennon avulla. Peli päättyy kun kymmen punaista pistettä on kerätty ja punainen hahmo voittaa pelin (Idea Labs 2015, 11).

Kodu ei ole pelkkää objektien ohjelmointia alusta loppuun, vaan Kodussa voidaan myös muokata olemassa olevaa maailmaa ulkonäöllisesti (kuva 4).



Kuva 4. Maailman muokkaaminen.

Alueesta voidaan tehdä hyvinkin suuri, jolloin tietokoneelta vaaditaan tarvittavaa suorituskkyä pelin suorittamiseen. Maailman luonnissa voidaan muokata vuoristojen kokoja ja rakenteita, tai erilaisia maatyyppejä kuten vaikkapa vettä. Näin ollen maailmasta saadaan erittäin monimuotoinen ja värikäs (Idea Labs 2015, 21–22).

3 ALKEISOPAS JA SEN TESTAUS

Tässä oppinäytetyössä mukana tulevassa alkeisopetuksen ohjeessa (liite 1) käytiin läpi neljä perusasiaa Kodulla ohjelmoinnista. Ohjeessa käsiteltiin ohjelman käyttöönotto sekä asennus, ohjelman perustoiminnot työkaluineen, logiikan ja perusohjelmoinnin hahmotus, kaksinpelin toteutus, sekä yhden haastavamman pelin toteutus, jolloin nähdään ohjelman jatkokehityspotentiaali.

Ohje jaettiin neljälle eritasoiselle ohjaajalle kokeiltavaksi käytännön opetuksessa. Yksi ohjaajista oli tehnyt Kodun kanssa ohjelmointia jo vuodesta 2010 lähtien ja omasi aikaisempaa ohjelmointitaitoa. Yksi ohjaajista oli aloittelija, jonka kokemus kattoi tietokoneen peruskäytön ja tietojenkäsittelyn taidot, kuten Excelin tai Wordin. Kaksi muuta ohjaajaa olivat tältä väliltä ja olivat harjaantuneempia tietokoneen käytössä. Vain yksi testihenkilöstä omasi ohjelmointitaitoa.

Lisäksi peruskoulun opettajia yritettiin saada ohjeen testauksen piiriin. Tästä kerrottiin kolmelle eri koulun opettajalle yhteistyön toivossa. Yrityksistä huolimatta testausta heidän kanssaan ei voitu toteuttaa, koska kenelläkään ei ollut ylimääräistä aikaa tai mielenkiintoa ohjelman opettelemiseen, saati edes tuntien seurantaan. Käytännössä kaikki ohjaaminen ja valvominen jäi neljän testihenkilön varaan.

3.1 Oppaan testaaminen ohjaajilla ja testauksen tulokset

Jokaista ohjaajaa haastateltiin vapaasti ohjeen osilta ja pyydettiin kommentoimaan opasta jatkokehitystä varten. Pääasiassa puhuttiin ohjeiden ymmärrettävyydestä ja käytännön kokemuksista. Haastattelut suoritettiin erikseen avoimena kyselynä huomioiden kyseisen henkilön kommentit, kehitysideat ja ongelmat. Näin saatiin erilaisia näkökulmia jokaisen taitotasoon nähden.

Päädyin avoimeen haastattelumenetelmään nettikyselyn tai valmiin lomakkeen sijasta, koska pidin valmista kyselyä lomakkeineen tehottomana. Lomakkeisiin henkilöt eivät jaks välttämättä kirjoittaa pidempiä kommentteja tai mielipiteitä. Avoimena haastatteluna sain ihmiset enemmän miettimään ja näin ollen täydentämään tai esittämään jatkokysymyksiä.

Ohjaajat pääasiassa kehuivat ohjeen helppoa seurattavuutta. Asiat käytiin läpi vaihe vaiheelta yksitellen, jolloin erilaisten työkalujen merkitys avautui selvemmin myöhemmissäkin vaiheissa. Asia mitä kaivattiin enemmän, oli syvempi perehtyminen työkaluihin ja niiden mahdollisuuksiin haastavimmissa tilanteissa.

Toinen ongelma asioiden opetuksessa ja käyttöönotossa oli kielelliset ongelmat. Ohjelma on englanninkielinen ja opetuskäytössä se saattaa aiheuttaa vaikeuksia alakoulutasolla, sekä nuoremmilla käyttäjillä. Tästä aiheesta on myös keskusteltu Microsoftin kanssa, olisiko mitään mahdollisuuksia ohjelman käännökselle koska ohjelma löytyy kuitenkin usealla kielellä. Asiasta oltiin myös yhteydessä Iso-Britannian Kodu-yhteisöön ja opettajiin jotka kertoivat, että käännöstyö on aina yksilöllistä ja se toteutetaan itse muokkaamalla ohjelman XML-tiedostoja.

Opetuksessa ohje on toiminut mainiosti ja sitä hyödynnettiin kokeiluna myös itse oppilaiden käytössä monisteina. Tässä ryhmässä oppilaat olivat alakoulun viidesluokkalaisia. Tällä testattiin, osaavatko oppilaat itse hyödyntää kyseistä liitettä käytännössä. Tämä osoittautui osin onnistuneeksi, mutta osa oppilaista ei jaksanut lukea tai seurata ohjeita, jolloin tarvittiin ohjaajan tukea.

3.2 Kolme kehitysideaa

Haastattelun ja testauksen tuloksista nostettiin esiin kolme tulevaisuuden kehitysideaa. Ensimmäinen on ohjelman kääntäminen, koska ohjelmaa löytyy myös eri kielillä. Microsoft ei kyselyiden mukaan tyrmää ehdotusta käännöstyöstä, mutta kertoo että se toteutetaan aina itse ilman heitä. Microsoft vain viimeiseksi tarkistaa ohjelman käännöksen oikeellisuuden ja näin ollen hyväksyy tai hylkää

sen. Tutkisin myös verkosta saatavia ohjelmia käännöstyön toteuttamiseen, hyvältä vaikuttava olisi Lingobit Localizer-ohjelma, jota myös Iso-Britanniasta suositeltiin. Ohjelman maksullisuudesta johtuen en päässyt itse testaamaan ohjelmaa käytännössä. Käännös osoittautuu tärkeäksi jos Kodun opetusta halutaan lisätä jatkossa muihinkin kouluihin ja halutaan opetuksen piiriin nuorempia käyttäjiä.

Toisena kehitysideana olisi haastavamman oppaan valmistaminen jatko-opetukseen tai yläkouluihin. Opetuksen suosiosta johtuen ja opetustunteja seuratesa, osalla oppilaista oli luontaisia kykyjä ohjelmoinnin oppimisessa. Tätä varten voitaisiin kehittää jatko-opas, jossa käydään läpi ohjelman monimutkaisemmat muuttujat ja kuinka ohjelman algoritmi konnaisuudessaan toimii. Oppaan ideana ei olisi niinkään itse opas, vaan kurssin johtaminen. Oppaan ympärille voitaisiin täten rakentaa kurssi, jossa lopuksi oppilaat tekevät itse omaa peliä, hyödyntäen aikaisempia ohjeita ja verkosta saatavaa materiaalia. Tämän kaltaisella oppaalla saataisiin myös jatkumo ohjelman opettamiseen yläkoulussa, koska ohjelma olisi jo tullut tutuksi alkeiskurssistaan alakoulussa.

Kolmantena kehitysideana olisi koulujen kanssa käytävä yhteistyön suunnittelu. Kodua voitaisiin hyödyntää perusaineiden opetuksessa, ei pelkästään ohjelmoinnin opetuksessa. Esimerkiksi matematiikka voi olla kouluaineena monelle erittäin haastavaa ja Kodu voisi olla ratkaisu tähän ongelmaan. Kodulla voitaisiin rakentaa leikinomaisesti oppitunti, jossa käydään joitakin matemaattisia tehtäviä läpi hyödyntäen Kodua. Tehdään esimerkiksi tehtävä, jossa käyttäjän tarvitsee ratkaista laskutehtävä pelissä päästäkseen eteenpäin seuraavalle tasolle. Näin ollen lapset voisivat olla motivoituneempia aineen opiskeluun ja samalla myös oppisivat matematiikkaa kuin vahingossa.

4 YHTEENVETO

Työn tavoitteena oli saada itse tuotettu ohje käytäntöön opetustilanteissa, hyödyntäen ulkopuolisia ohjaajia ja näin ollen mahdollisti omien päättelyiden ja hyväksi havaitsemieni opetustapojen seuraamisen käytännössä muiden toteuttamana. Työskentelymenetelmänä on toiminut Kodun avulla opittu tieto ja taito, joita olen hyödyntänyt käytännön opetustyössä. Ohjelma on vielä kokeiluasteella Suomessa eikä tutkimustietoa ole saatavilla. Ohje tehtiin näin ollen omien kokemusten pohjalta.

Tulokset kuitenkin osoittavat käytetyn oppaan toimivaksi opetuksessa, mutta riittämättömäksi syventävään opetukseen ylemmillä opetusasteilla. Englannin kieli saattoi muodostua ongelmaksi nuoremmilla opiskelijoilla, joka saattoi johtaa mm. turhautumiseen. Syventävän ohjeen tai kurssin suunnittelu tulisi toteuttaa käymällä läpi lähemmin ohjelman algoritmeja ja ohjelmointirakennetta. Näiden pohjalta voitaisiin valmistaa haastavampi ohje tai kurssi yläkoulun käyttöön ja jatkumoa opetukselle.

Ensimmäisenä jatkokehitystä voitaisiin tehdä ohjelman kääntämisessä suomeksi. Vaadittavien XML-tiedostojen editointia voitaisiin helpottaa apuohjelmia käyttäen. Toiseksi aihe tarjoaa jatkotyöskentelymahdollisuuksia syventävän materiaalin tuottoon yläkouluikäisten opetukseen. Näin kyetään vaativimpien pelien suunnitteluun. Kolmanneksi tulisi suunnitella opetusalan ammattilaisten kanssa Kodun hyödyntämistä tuntiopetuksessa. Esimerkiksi matematiikka tai muut luonnontieteet.

LÄHTEET

Mykänen, J. & Liukas, L. 2014. Koodi2016. Helsinki: Lönnberg Print.

Evans, C. 2013. Kodu: Microsoft's Tool for Schools – An Interview with Stuart Ball. Viitattu 21.2.2015 <http://thereticule.com/kodu-microsofts-tool-for-schools-an-interview-with-stuart-ball/>.

Kodu 2015. What is Kodu? Viitattu 3.1.2015 <http://www.kodugamelab.com/about/>.

Idea Labs 2015. Designing games with Kodu. Viitattu 20.1.2015
<http://media.planetkodu.com/workshop/Designing%20Games%20with%20Kodu%20Game%20Lab%20-%20Participant%20Manual%20v2.pdf>.

Kelly, J. 2013. Kodu for Kids: The Official Guide to Creating Your Own Video Games. Indiana: Que

Price, E. 2014. Beginning Microsoft Small Basic: Chapter 1: Introducing Small Basic.
<http://social.technet.microsoft.com/wiki/contents/articles/16394.beginning-microsoft-small-basic-chapter-1-introducing-small-basic.aspx>.

Alexi Myllymäki

OPAS KODUN ALKEISIIN

Sisältö

1 KODUN ASENNUS JA SIIHEN LIITTYVÄT TOIMENPITEET	3
Asennus ja käyttöönotto	3
2 OPETUKSEN TOTEUTUS	8
2.1 Ensimmäinen sessio. Ohjelman perustyökalut	8
2.2 Toinen sessio. Peruslogiikan hahmotus.	17
2.3 Kolmas sessio. Kaksinpeli	25
2.4 Neljäs sessio. Haastavampi peli.	33
3 MITÄ ALKEIDEN JÄLKEEN?	40

1 KODUN ASENNUS JA SIIHEN LIITTYVÄT TOIMENPITEET

Ensimmäisen osion tarkoituksena on käydä läpi Kodun asennus ja käyttöönotto. Asennuksessa käydään vaiheittain läpi kaikki mitä käyttäjän tarvitsee tehdä ja tietää, että Kodu saadaan toimimaan haluamalla tavalla omalla tietokoneella. Tällä hetkellä minimi vaatimukset ohjelmalle ovat seuraavat:

- Windows XP, Windows Vista, Windows 7, Windows 8/8.1
- DirectX 9.0c sekä Shader Model 2.0:laa tukeva näytönohjain.
- .Net Framework 3.5 ja XNA Framework 3.1

Asennus ja käyttöönotto

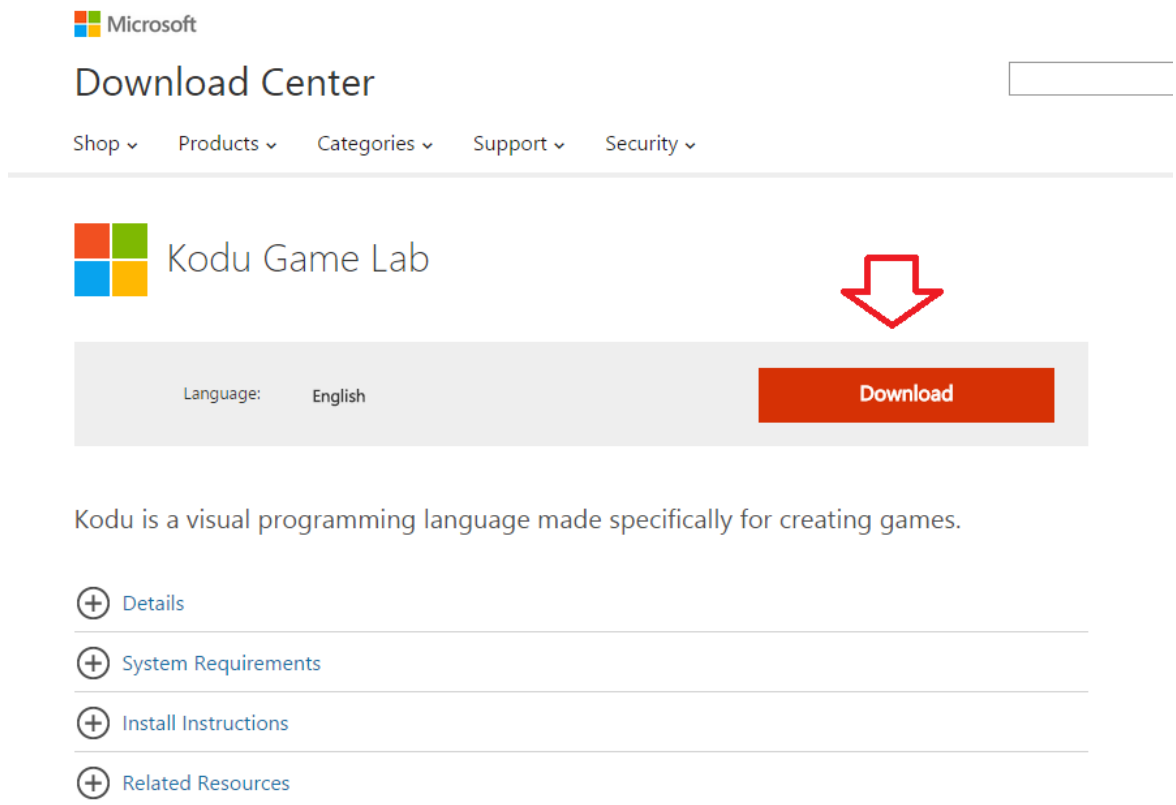
Kodun pystyy tällä hetkellä asentamaan ilmaiseksi Windows käyttöjärjestelmälliseen tietokoneeseen tai tablettiin. Tietokoneessa pitää olla DirectX:ää tukeva näytönohjain jossa on asennettuna DirectX 9.0c. Näiden lisäksi tarvitaan vielä NET Framework 3.5 tai uudempi ja XNA Framework 3.1 tai uudempi. Kaikki tarvittava löytyy Microsoftin kotisivuilta tarvittaessa jos niitä ei kyseisessä tietokoneessa jo ole.

Huomioitavaa on ero asennuksissa eri käyttöjärjestelmien välillä. Tässä ohjeessa käydään läpi perusteellisemmin asennus Windows 7 käyttöjärjestelmään. Kodu löytyy myös Windows 8:lle, mutta ladatakseen käyttäjällä tarvitsee olla Microsoft tili ja ohjelma ladataan Microsoftin omasta kaupasta ilmaiseksi. Vanhemmilla käyttöjärjestelmillä lataus tapahtuu osoitteesta www.kodugamelab.com. Tällä hetkellä mainitsemasta nettiosoitteesta ladattava versio ei toimi Windows 8:lla.



Kuva 1. Kodun lataus painike www.kodugamelab.com sivustolla.

Kuvassa (Kuva 1) selaimella on menty osoitteeseen www.kodugamelab.com ja heti ensimmäisellä sivulla klikataan vasemmassa yläkulmassa olevaa "Get Kodu" linkkiä. Linkin avaamisen jälkeen selain siirtää käyttäjän suoraan Microsoftin sivuille lataamaan ohjelman (Kuva 2).



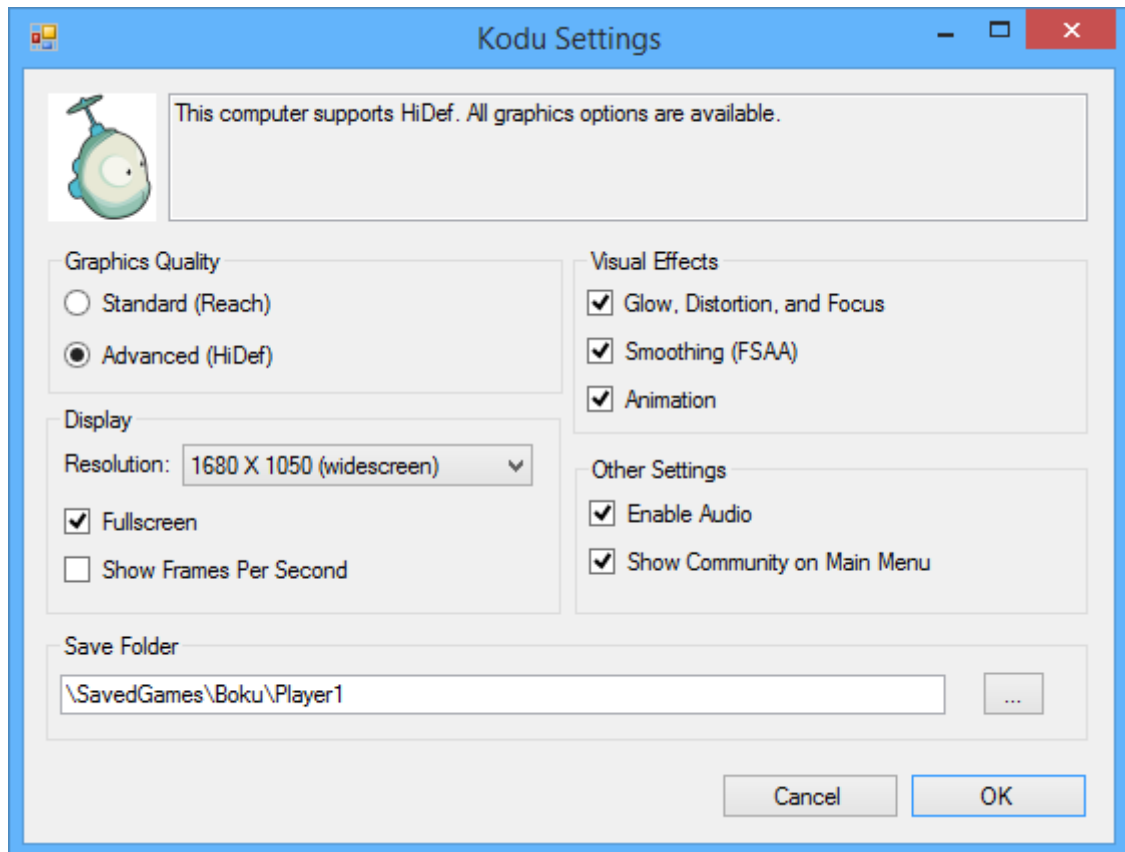
Kuva 2. Microsoftin kotisivuilla oleva lataus painike.

Latauksen alkaessa kysytään sijaintia minne tiedosto ladataan. Lataamisen jälkeen mennään haluttuun sijantiin ja asennetaan ohjelma tuplaklikkaamalla kyseistä tiedostoa.



Kuva 3. Asennus tarkastaa tarvittavat komponentit ja ilmoittaa jos niitä ei ole asennettu.

Asentaminen on hyvin suoraviivaista ja yksinkertaista. Tällä tarkoitetaan vaihe vaiheelta tapahtuvaa asennusta joista ensimmäinen on ladatun tiedoston avaaminen heti koneelle lataamisen jälkeen. Asennusohjelma tarkistaa onko tarvittavat komponentit asennettuna ja jos ei ole, niin asennusohjelma tarjoaa linkkiä niiden lataamiseen suoraan Microsoftin sivuilta (Kuva 3). Muuten asennusohjelma kysyy tallennettavaa polkua ohjelmalle ja kuvakkeiden lisäämistä työpöydälle. Asennuksen suorittamisen jälkeen on suositeltavaa tarkistaa ohjelman asetukset kohdasta "Kodu Settings" joka yleensä löytyy työpöydältä Kodun käynnistämiskuvakkeen vierestä tai aloitusvalikosta nimellä "Configure Kodu Game Lab". Seuraavasta kuvasta näkyy esimerkki asetusikkunasta (Kuva 4).



Kuva 4. Asetusten muuttaminen.

Jos käyttäjällä ei ole tietoa mitä tehdä asetuksissa, niin helpoin vaihtoehto on valita "standard" asetus jolloin ohjelmaa asettaa yleisimmät asetukset. Nämä yleisesti toimivat valtaosassa nykypäivän tietokoneista. Jos tiedetään että koneessa on enemmän suorituskkyä, niin voidaan myös valita "advanced" asetus jolla päästään käsiksi parempiin grafiikka-asetuksiin. Asetusten jälkeen voidaan käynnistää itse ohjelma ja katsoa alkuvideon jälkeen, että kuinka ohjelma toimii. Jos heti alussa ohjelma vaikuttaa hitaalta tai ns. nykivältä, kannattaa mennä jälleen asetuksiin ja valita "advanced" vaihtoehto, jonka jälkeen voidaan tiputtaa vaikka resoluutiota ja yrittää uudelleen. Tätä voidaan jatkaa kunnes ohjelma toimii halutulla tavalla.

2 OPETUKSEN TOTEUTUS

Tässä luvussa käydään läpi ohjeita neljälle kerralle joita olen itse toteuttanut. Jokaiselle opetus keralle voidaan halutessa tehdä suunnitelma tai käyttää sitten näitä esimerkkejä joita olen myös kokeillut ja viimeiseksi otetaan vielä haastavamman pelin ohjeet. Jokaisen opetuskerran aika on noin puolitoistatuntia.

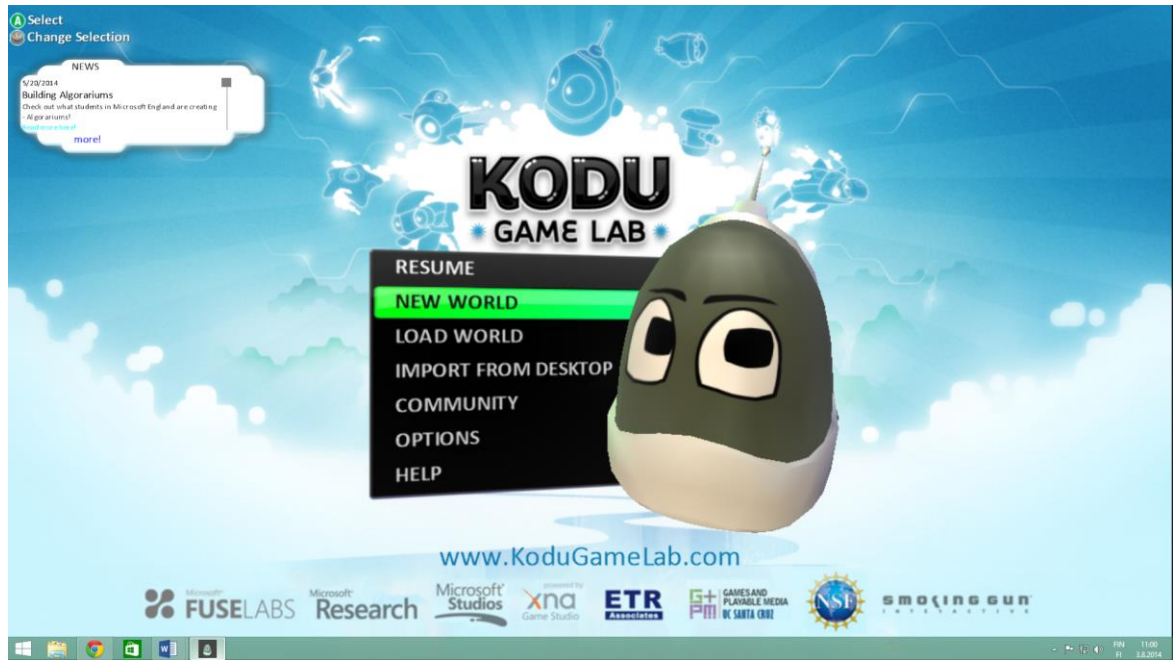
2.1 Ensimmäinen sessio. Ohjelman perustyökalut

Ensimmäisellä kerralla tehdään perusasioita ja opetellaan ohjelman käyttöä yleisellä tasolla. Oppilaat saavat itse tutustua ohjelmaan ja sen työkaluihin. Opetus on lähinnä maailman ja objektien luomista.

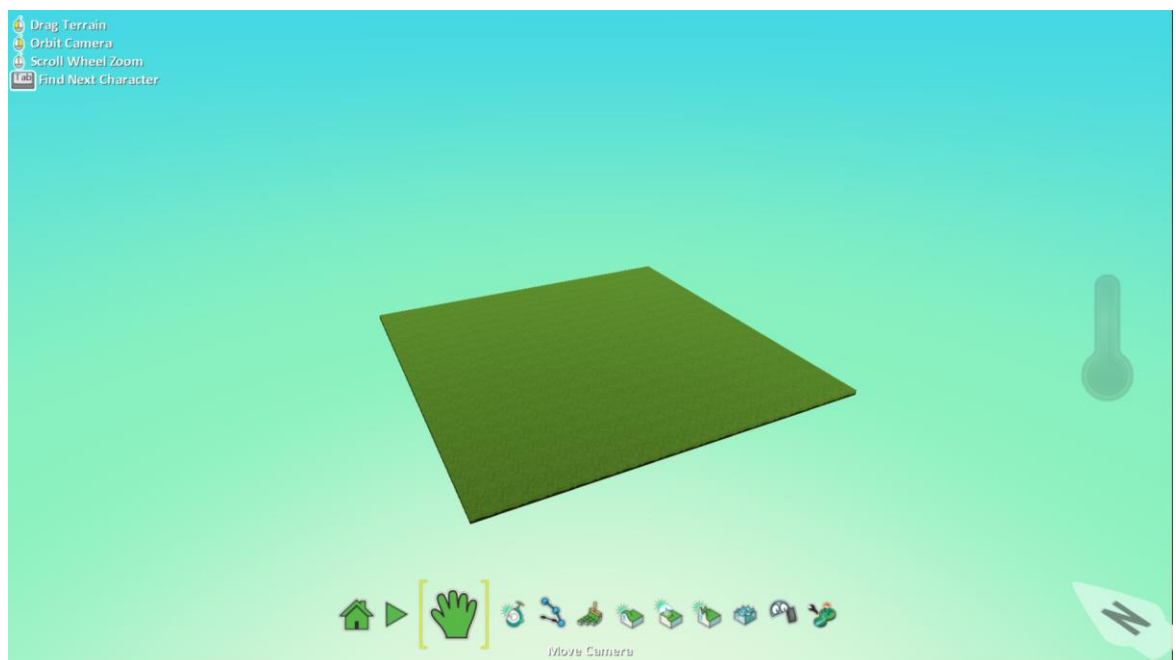
Kaikki alkaa siitä että ensimmäiseksi avataan Kodu ohjelma joko työpöydältä Windowsissa tai aloitusvalikon kautta. Jos käytössä on kosketusversio tai Windows 8/8.1, niin ohjelma voidaan aloittaa myös metro näkymästä jos ohjelma on siirretty sinne valmiiksi. Tämän jälkeen valitaan valikosta "New world" eli uusi maailma. Näin aukeaa uusi taso muokattavaksi ja tehtäväksi (Kuva 6 & 7).



Kuva 5. Työpöytä ikoni.



Kuva 6. Kodun aloitusvalikko ohjelman käynnistyttyä.



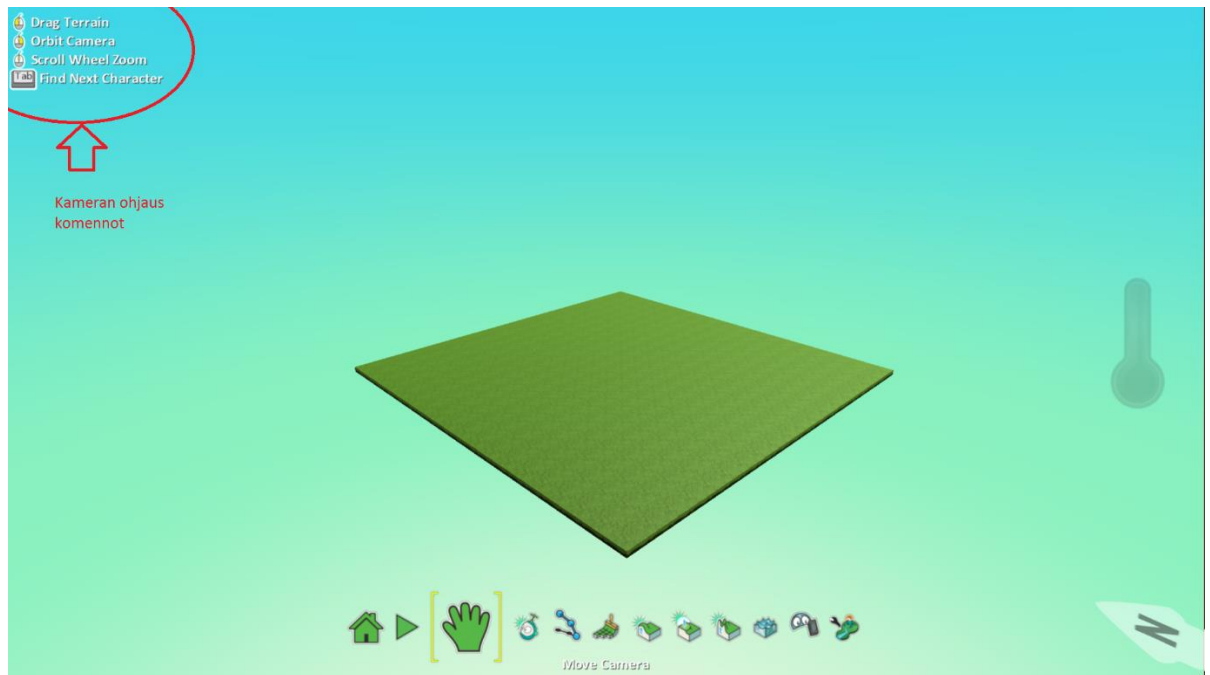
Kuva 7. Uusi maailma.

Kun uusi maailma on avattu, niin ensimmäiseksi voidaan käydä läpi alhaalla olevassa valikossa olevia ikoneita ja niiden toimintoja läpi. Ensimmäisellä keral-

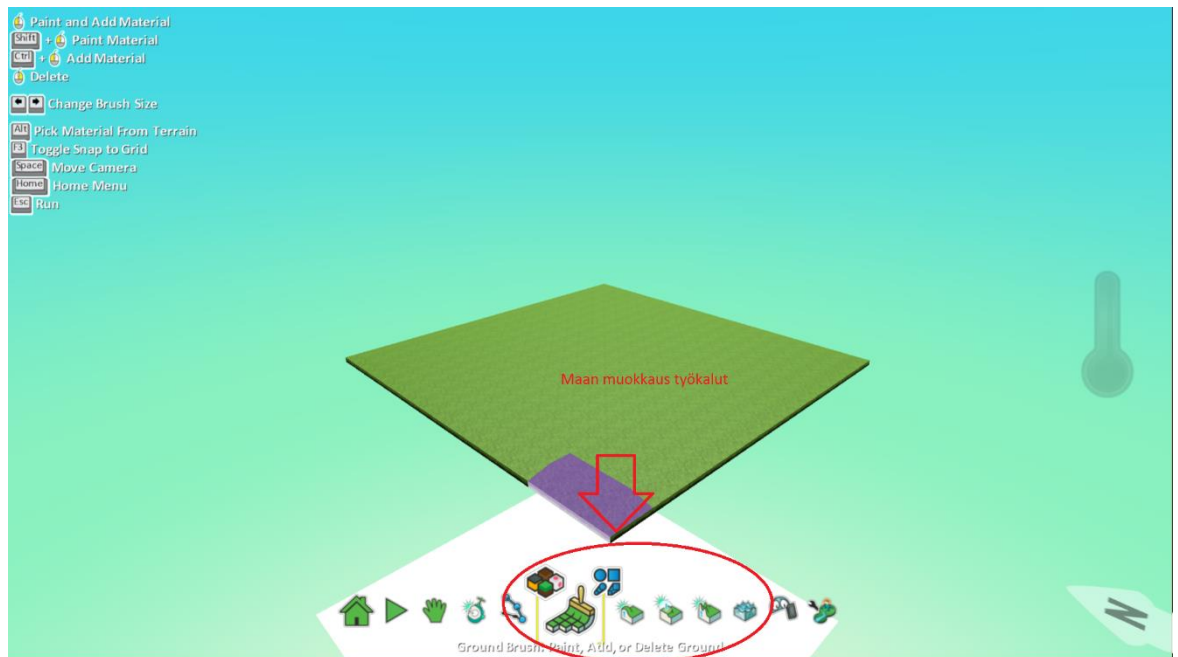
la tärkeimpiä työkaluja ovat maailman muokkaus työkalut ja kameran ohjaus. Näihin kuuluu maan materiaali, vesi ja vuori työkalut. Kameran ohjaus tapahtuu hiiren ja näppäimistön yhteistyöllä. Muita toiminnot alarivissä ovat objektien luonti, polun rakennus, poisto ja maailman asetukset.



1. Koti (Tällä pääsee takaisin päävalikkoon jossa voidaan mm. lopettaa ohjelma, tallentaa tai ladata peli.)
2. Pelaa (Tätä painamalla päästään pelaamaan luotua maailmaa ja näin testamaan miten toiminnot toimii)
3. Käsi/kamera (Tällä työkalulla liikutetaan kameraa ja valitsemalla tämä päästään näkemään myös oikeassa yläkulmassa olevia kameran liikuttamiseen tarkoitettuja ohjeita.)
4. Objektit (Objektityökalulla voidaan luoda objekteja maailmaan ja hallita niitä)
5. Polku työkalu (Tällä työkalulla voidaan luoda polkuja ja ratoja, mitä pitkin esimerkiksi voidaan määrittää tietokoneohjattujen hahmojen kulku)
- 6-10. Maailman luontiin tarkoitettuja työkaluja, selitetään maailman luonnissa seuraavassa luvussa.
11. Kumitus (Tällä voidaan poistaa kaikkea jo tehtyä maailmasta.)
12. Maailman asetukset (Täältä voidaan vaihtaa maailman asetuksia halutunlaiseksi, esimerkiksi taivaan väriä, kameran oletus asetuksia, laskureita jne.)



Kuva 8. Kameran ohjaukseen tarkoitetut toiminnot.



Kuva 9. Maan muokkaukseen tarkoitettuja työkaluja.



Kuva 10, 11, 12. Maailman luontiin tarkoitetut työkalut.

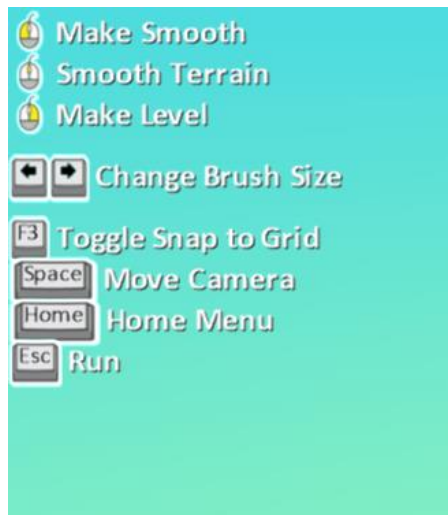
Ensiksi voidaan ottaa käsittelyyn työkalut maailman luomiseen

Kuva 10. Pensseli, materiaali ja muoto työkalut. Näillä luodaan perustukset maailmalle joista tärkein on pensseli maan luonnissa. Materiaalilla valitaan maan materiaali ja ulkonäkö. Muoto kalulla voidaan valita erilaisia muotoja ja kokoja mitä pensselillä ”maalataan”.

Kuva 11. Vuori työkalut. Näillä luodaan vuoria ja muotoillaan maan korkeutta ja rakennetta. Ensimmäisellä vasemmalta nostetaan maata ja toisella tasoitetaan. Tasoituksella voidaan tasoittaa epätasaisuudet. Kolmannelta vasemmalta tehdään ns. piikikästä maata. Tällä voidaan tehdä rosoisia maa alueita ja muuta epätasaisuutta.

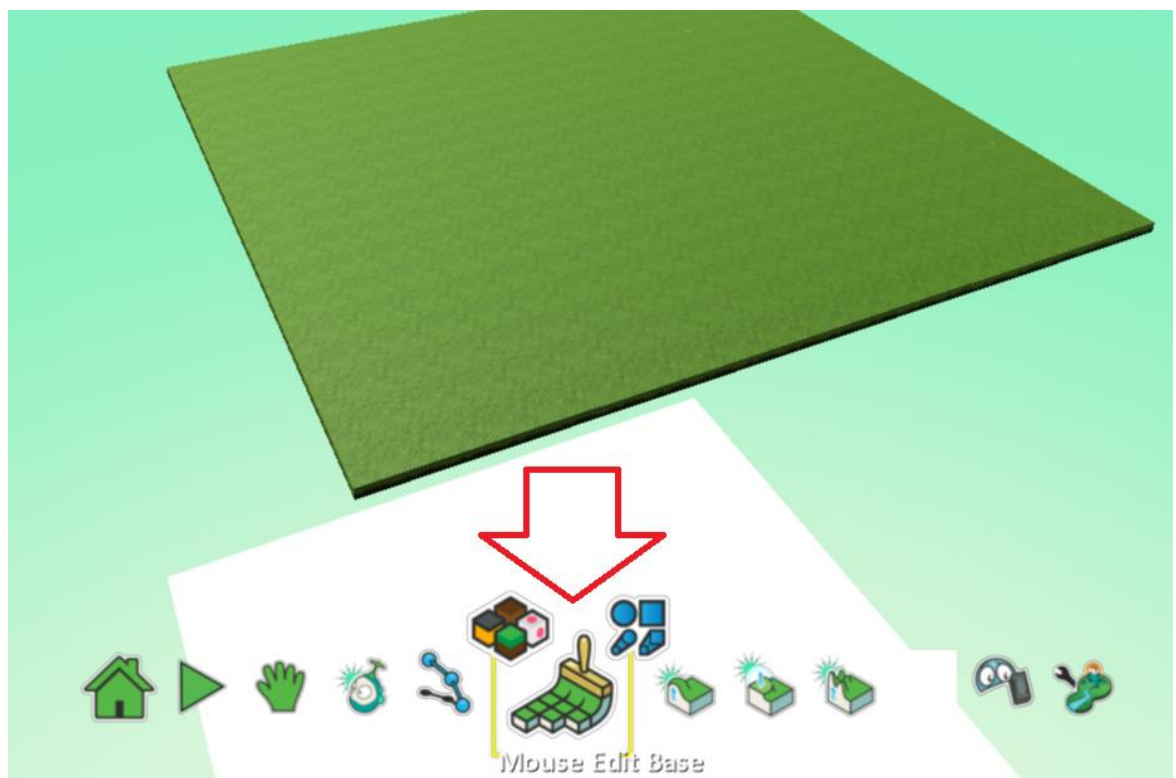
Kuva 12. Vesi työkalu. Vesi työkalulla luonnollisesti luodaan vesi alueita. Vesi työkalu toimii kun maata luodaan kuppimaiseksi alueeksi tai ns. kuopaksi. Kun vesityökalua sitten käytetään kuopassa, niin vesi nousee sen sisällä. Jos tällaista kuoppaa ei ole niin vesi valtaa koko alueen tai maailman.

Esimerkissä käytetään hiirtä sekä näppäimistöä. Kun hiiri viedään kuvakkeen päälle, ilmestyy sen viereen toisia pienempiä kuvakkeita. Klikkaamalla kuvakkeita, saadaan erilaisia vaihtoehtoja aina muodoista erilaisiin materiaaleihin kun muokataan maata halutun laiseksi. Myöskin kaikki komennot ja ohjeet siihen että mistä näppäimestä tapahtuu mitäkin, löytyy aina vasemmasta yläkulmasta (Kuva 13).

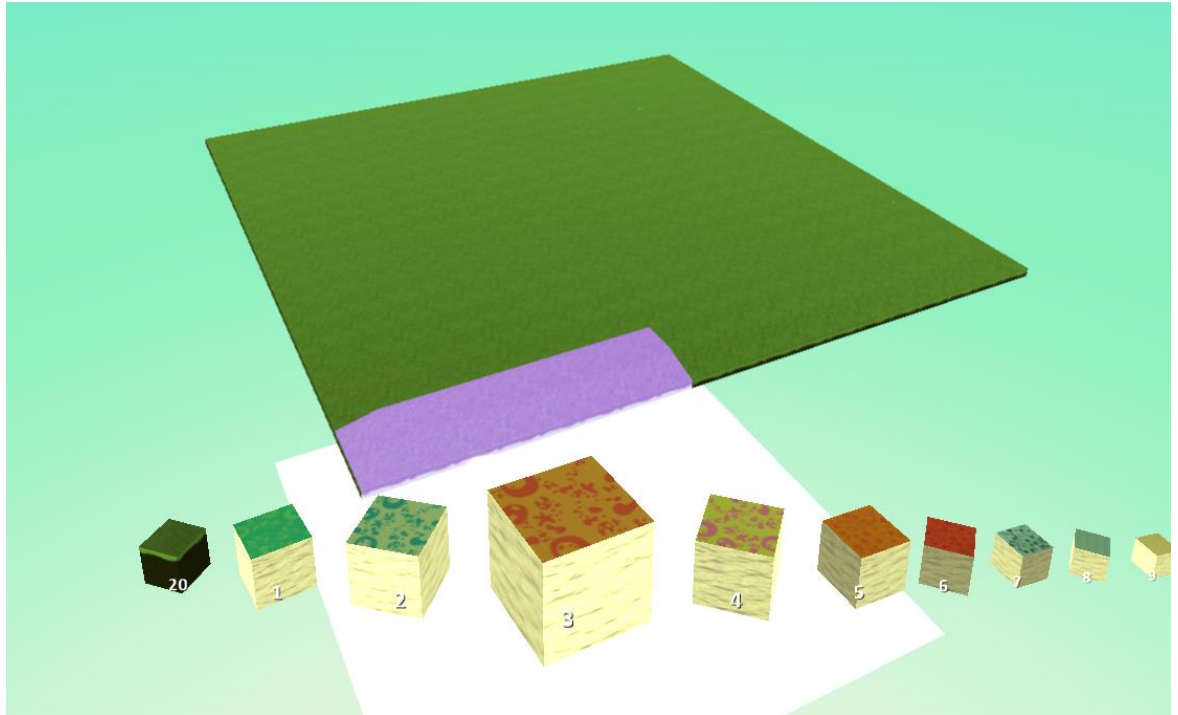


Kuva 13. Komennot näppäimille.

Aloitetaan tekemällä enemmän maata ja valitaan pensseli. Viereen ilmestyy myös materiaali ja muoto kalut (Kuva 14). Valitaan nyt kuitenkin vain materiaali ja sieltä haluttu materiaali.

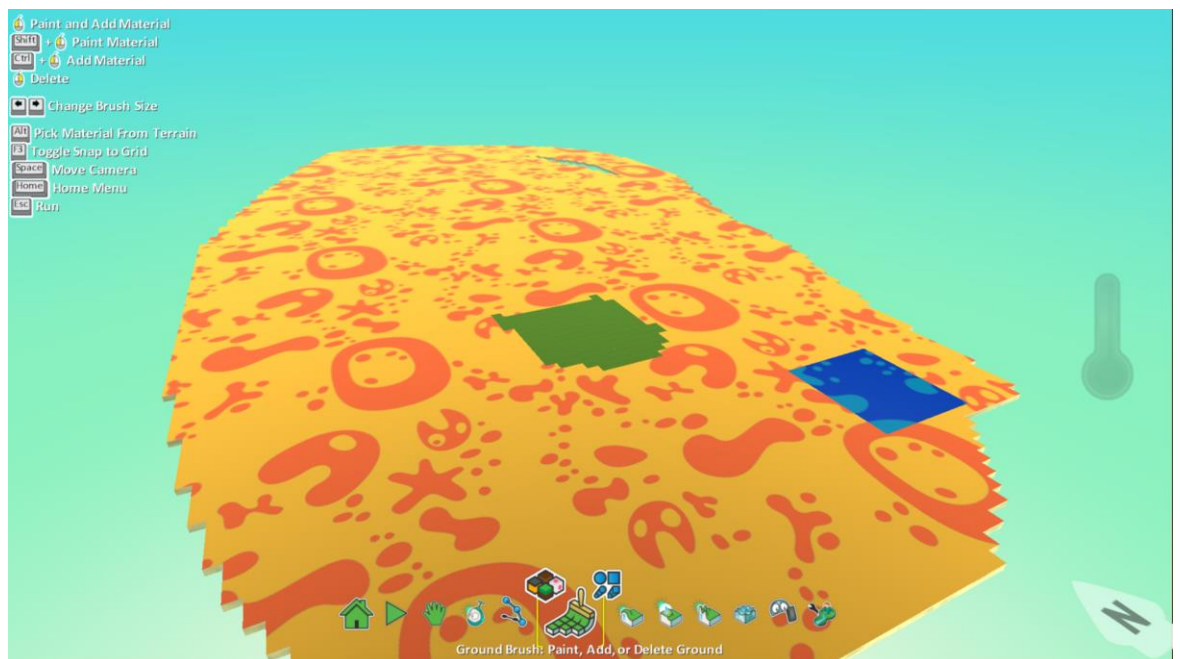


Kuva 14. Valitaan pensseli maan luomiseksi.



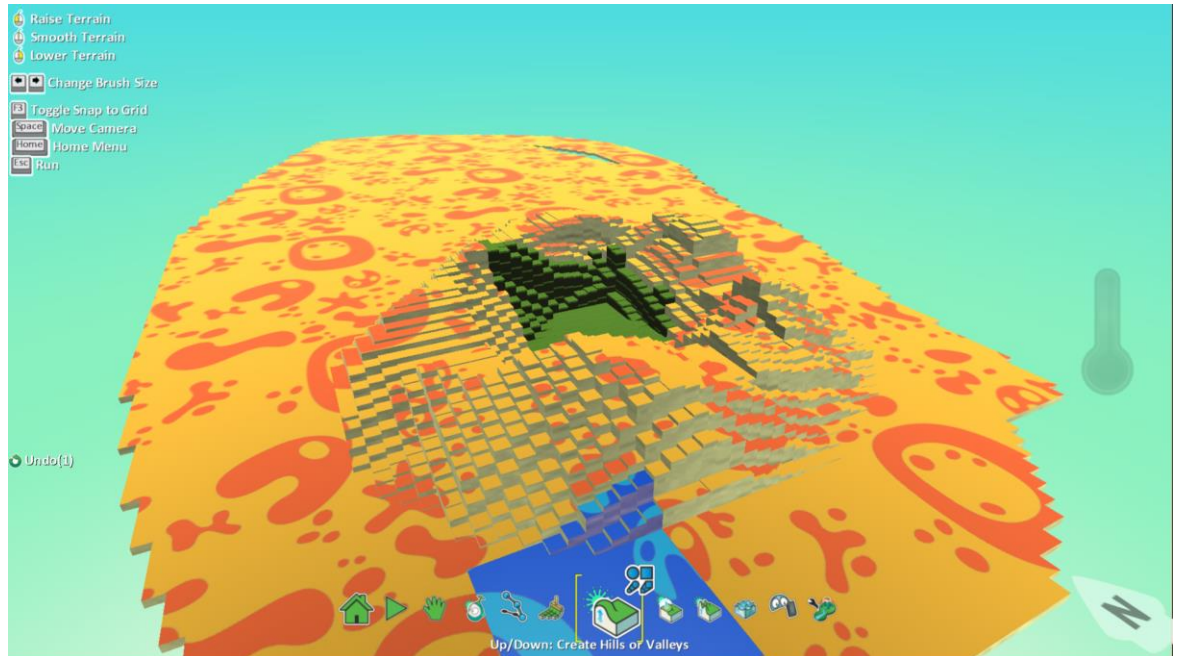
Kuva 15. Valitaan haluttu materiaali.

Tämän jälkeen sitten tehdään haluttu määrä maata, kuitenkin sen verran että saadaan seuraavassa vaiheessa tehtyä myös vuoria (Kuva 16).



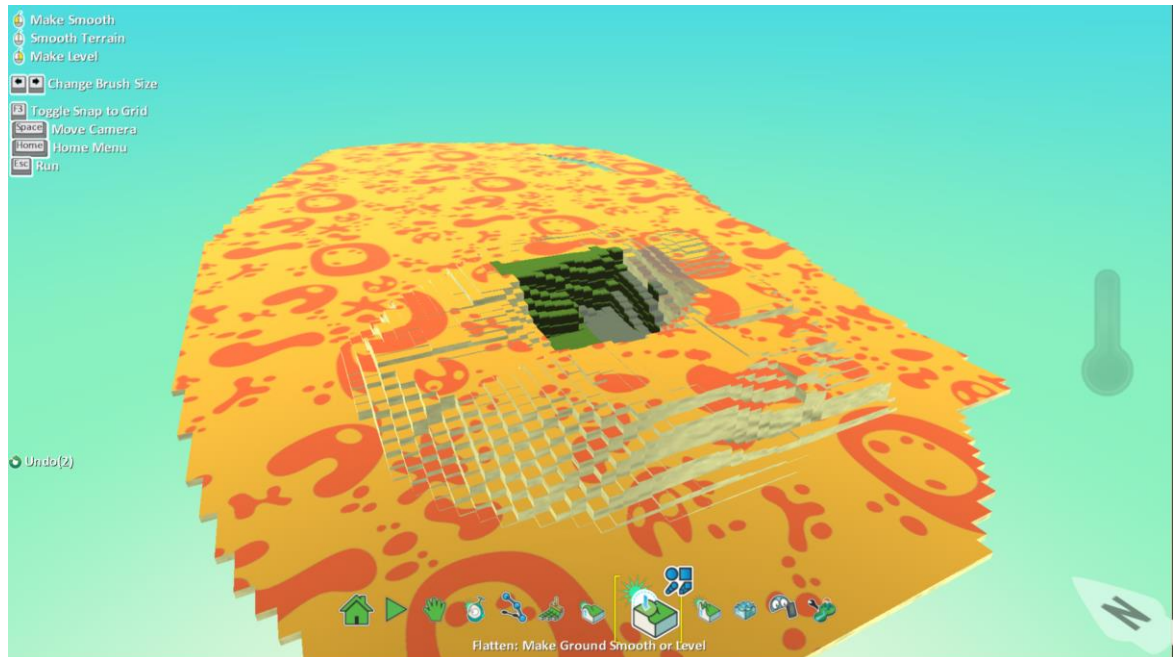
Kuva 16. Maa rakentelun jälkeen.

Tämän jälkeen voidaan tehdä vuoria ja tehdäänkin maahan sen verran vuorta kuopan aikaansaamiseksi, jolloin voidaan yrittää sen jälkeen luoda kuoppaan vettä. Eli valitaan perus vuori työkalu ja tehdään sillä (Kuva 17).



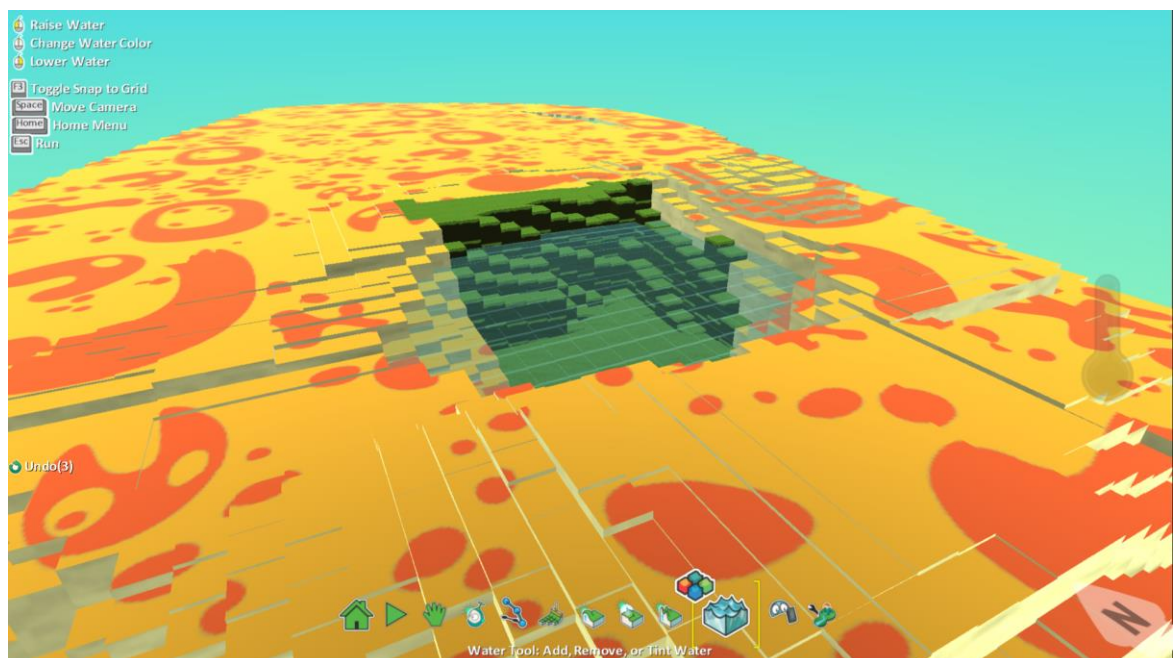
Kuva 17. Tehdään vuorista kuoppa.

Tämän jälkeen voidaan tasoittaa vuoria, että saadaan alueesta siistimmän näköinen. Valitaan seuraava työkalu vuorikalun vierestä ja tasoitetaan sillä (Kuva 18).



Kuva 18. Tasoitetaan aluetta.

Seuraavaksi valitaan vesityökalu ja viedään se juuri luodun kuopan keskelle ja luodaan vettä (Kuva 19).



Kuva 19. Veden luonti.

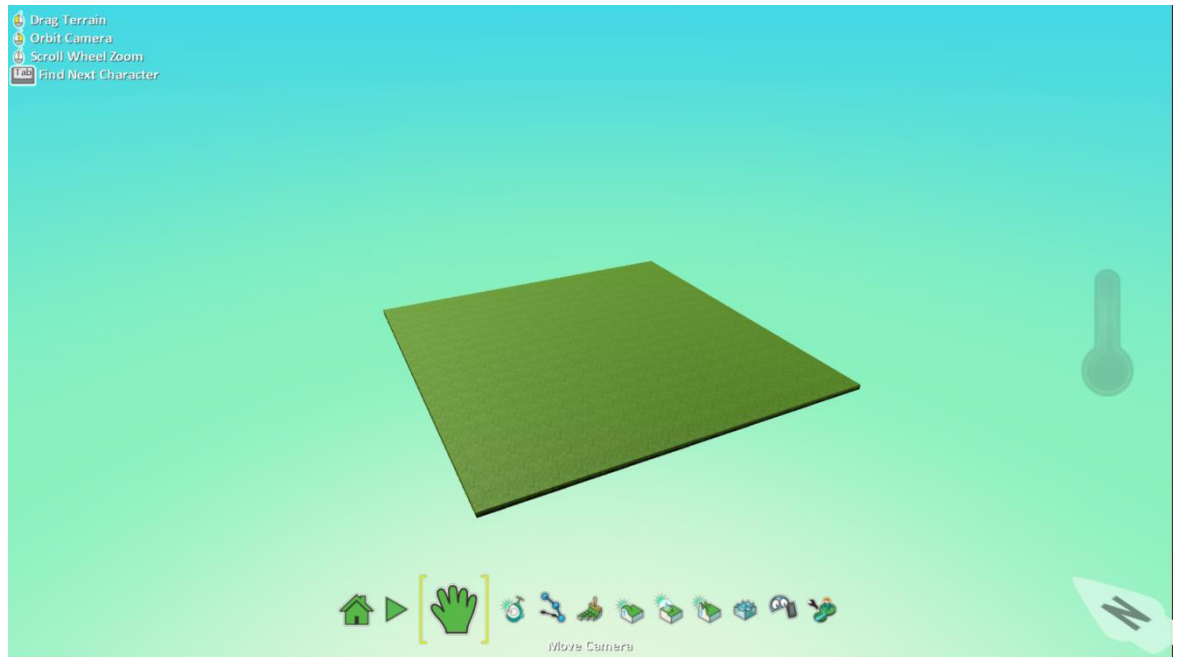
Ensimmäisen osion sisältö on tässä. Käyttäjän pitäisi hallita peruskalujen käyttö ja niitä hyödyntämällä rakentaa erilaisia maailmoita joita tarvitaan seuraavissa osioissa ja tulevaisuudessa kun aletaan tekemään isompaa peliä. Seuraavassa osiossa käydään läpi perus logiikan piirteitä ja tehdään ensimmäinen pienimuotoinen peli.

2.2 Toinen sessio. Peruslogiikan hahmotus.

Jokaisen session jälkeen voidaan uuden session alussa nopeasti kerrata vanhoja asioita läpi ja palauttaa mieleen jo tehtyjä tehtäviä. Toisen kerran tavoitteena on objektien luonti ja niiden perusohjelmointi. Perusohjelmoinnilla tarkoitetaan peruslogiikan hahmottamista.

Tällä kerralla voidaan joko jatkaa jo aloitettua maailmaa jota editoitiin ensimmäisellä kerralla tai aloittaa kokonaan uusi. Käytetään kuitenkin uutta maailmaa hahmotuksen helpottamiseen. Tehdään nyt peli jossa tarkoituksena on kerätä puusta putoavia omenoita.

Ensimmäiseksi aloitetaan uusi maailma alkuvalikosta jolloin päästään uuteen maailmaan. Tämän jälkeen kun uusi maailma on avattu, niin voidaan laajentaa maa-alaa jonkin verran ja valitaan objekteista ”Kodu” ja jokin puu.

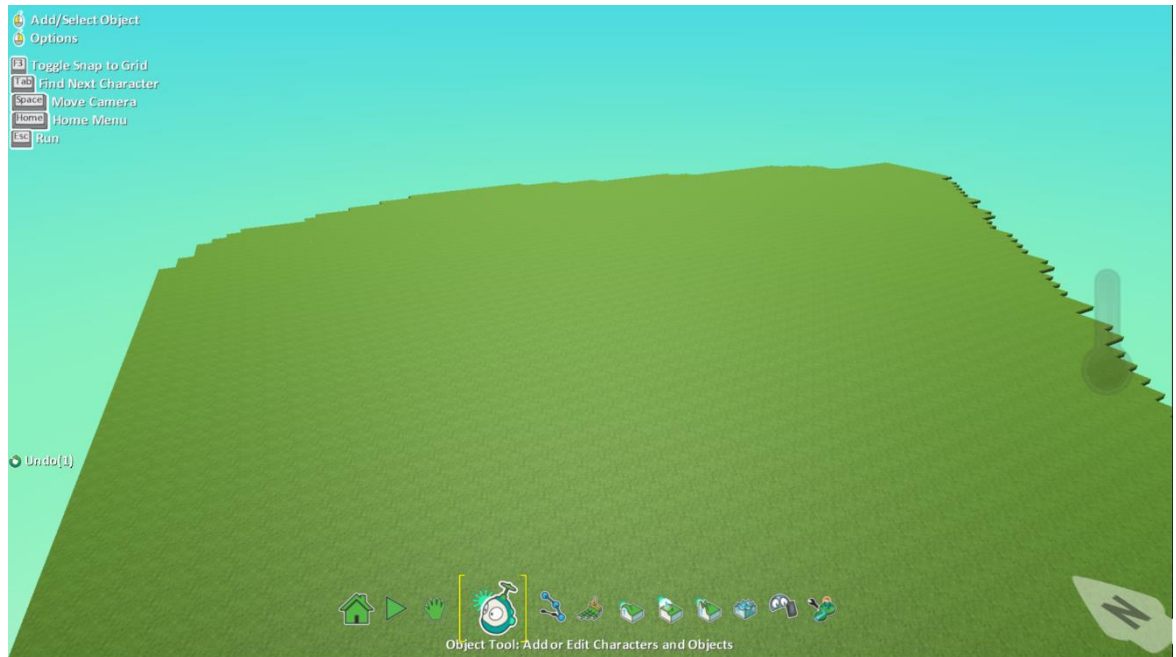


Kuva 14. Uusi maailma.



Kuva 15. Maata laajennettu ”pensseli”-työkalun avulla.

Itse objektien ja hahmojen luonti tapahtuu objekti työkalulla joka on valittuna seuraavassa kuvassa (Kuva 16). Objektin käyttö tapahtuu ensin ikonia klikkaamalla jonka jälkeen klikataan maailmassa kohtaa jonne halutaan jonkinlainen objekti luoda.

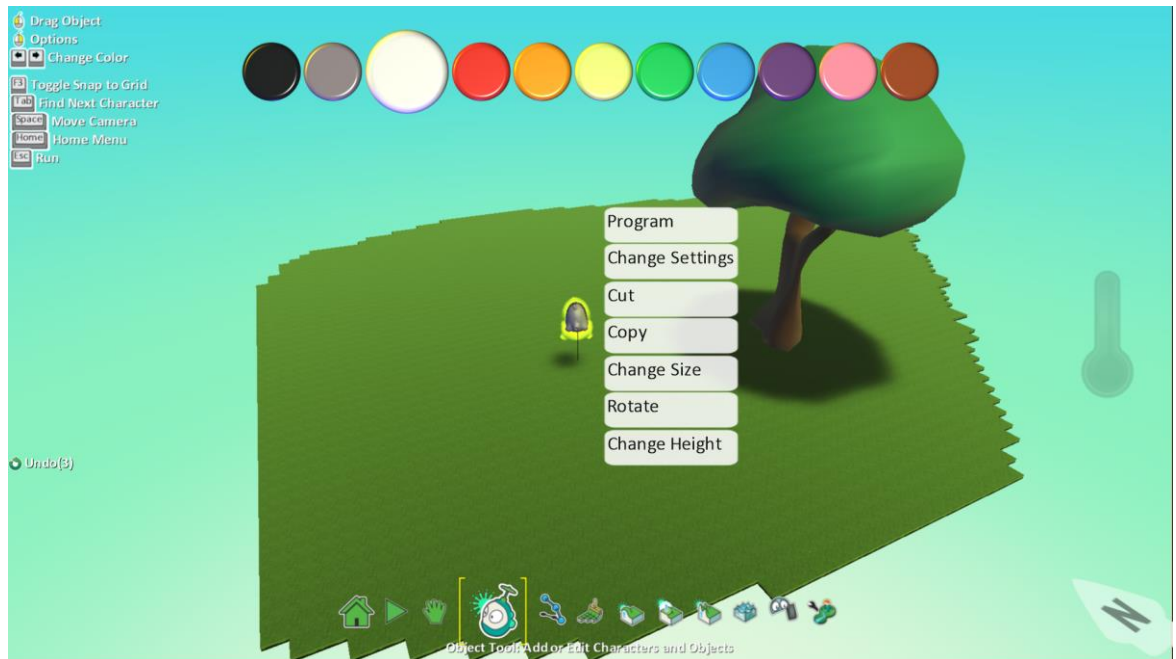


Kuva 16. Valitaan objekti työkalu.

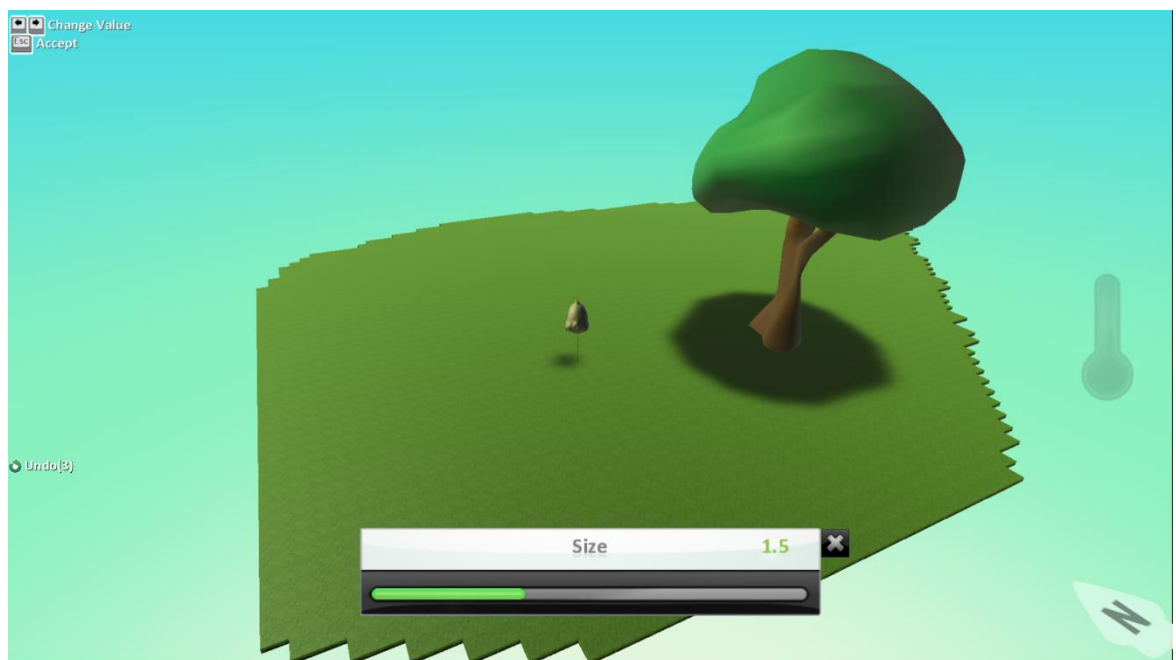


Kuva 17. Klikataan haluttua objektia jolloin se ilmestyy maailmaan.

Valitaan tällä kertaa Kodu ja sen jälkeen uudelleen jokin puu (Kuva 17).



Kuva 18. Valintoja oikealla hiiren näppäimellä.



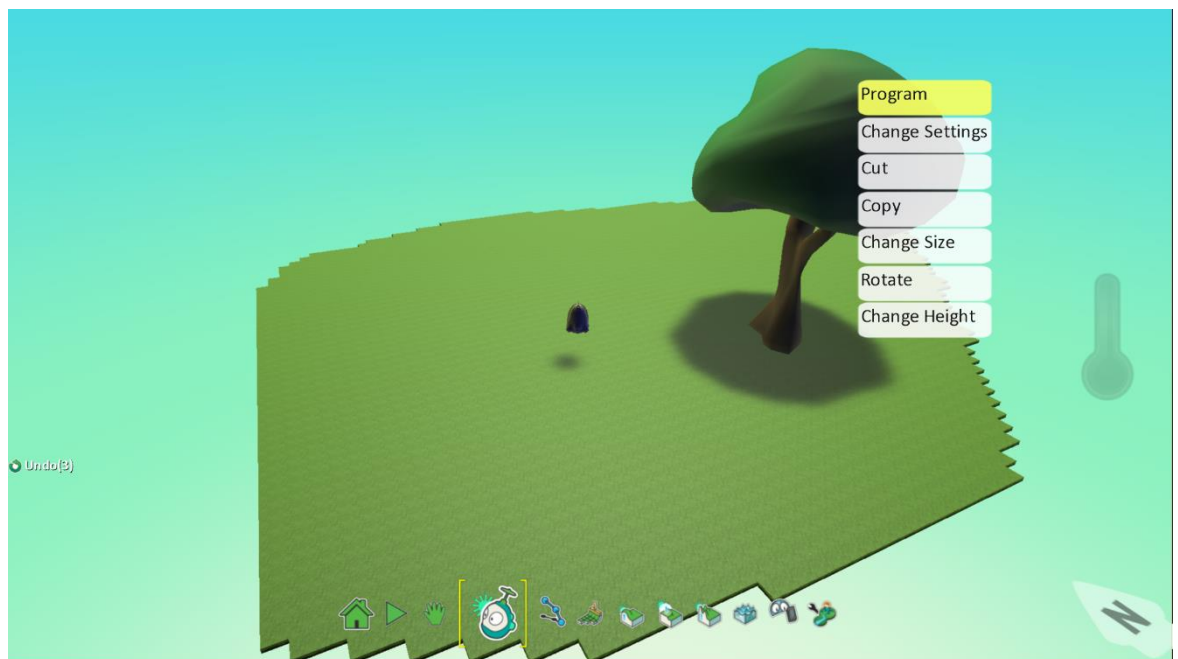
Kuva 19. Kodu ja puu.

Myös objektien värejä ja kokoja voidaan vaihtaa. Hiirtä käytettäessä kokoja voidaan vaihtaa oikeaa hiiren painiketta painaen objektin kohdalla jolloin aukeaa

eri vaihtoehtoja. Tässä tapauksessa olen muuttanut puun kokoa isommaksi (Kuva 19).

Ohjelmointi tapahtuu samalla tavalla, klikkaamalla oikeaa painiketta ja valitaan kohta "Program" eli ohjelmointi. Toisella opetuskerralla ohjelmoidaan puu luomaan omenoita joka viides sekunti ja ohjataan itse Kodu syömään niitä. Joka kerralla kun Kodu syö yhden omenan, pelaaja saa siitä yhden pisteen. Kun 5 pistettä on kerätty, peli päättyy.

Aloitetaan ensin puun ohjelmoinnista. Valitaan ensin "Program" valikko puusta johon aletaan rakentamaan sille vaadittuja tehtäviä (Kuva 20).



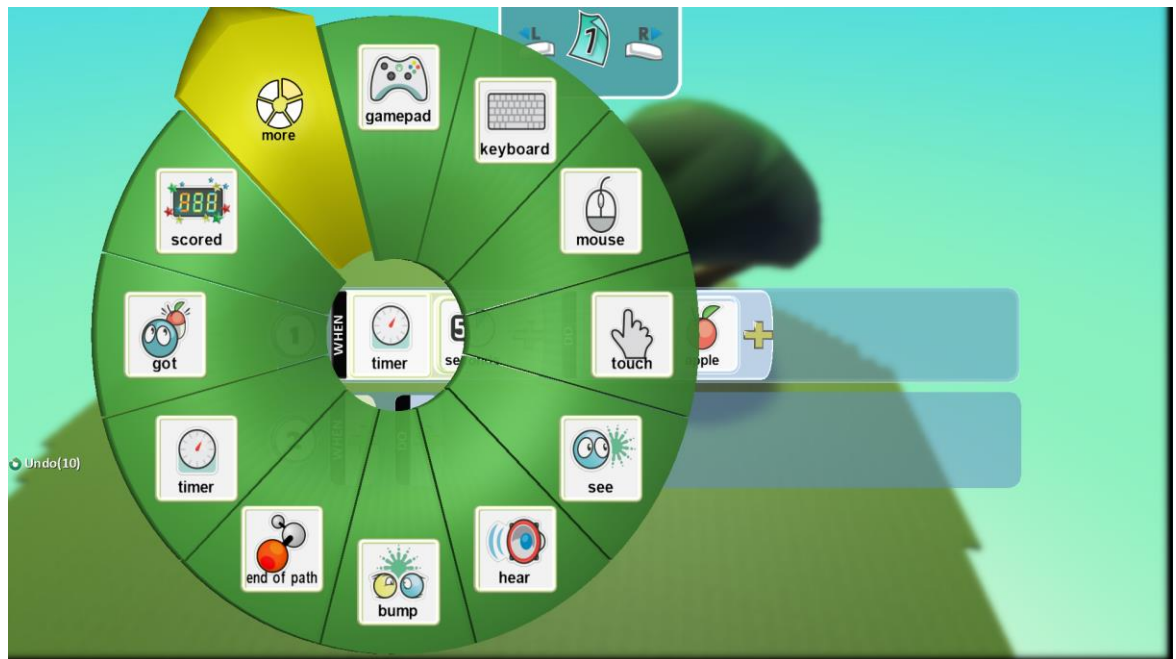
Kuva 20. Puun valikko josta valitaan ohjelmointi.

Valinnan jälkeen aukeaa ohjelmointi valikko jossa annetaan halutut toiminnot puulle.

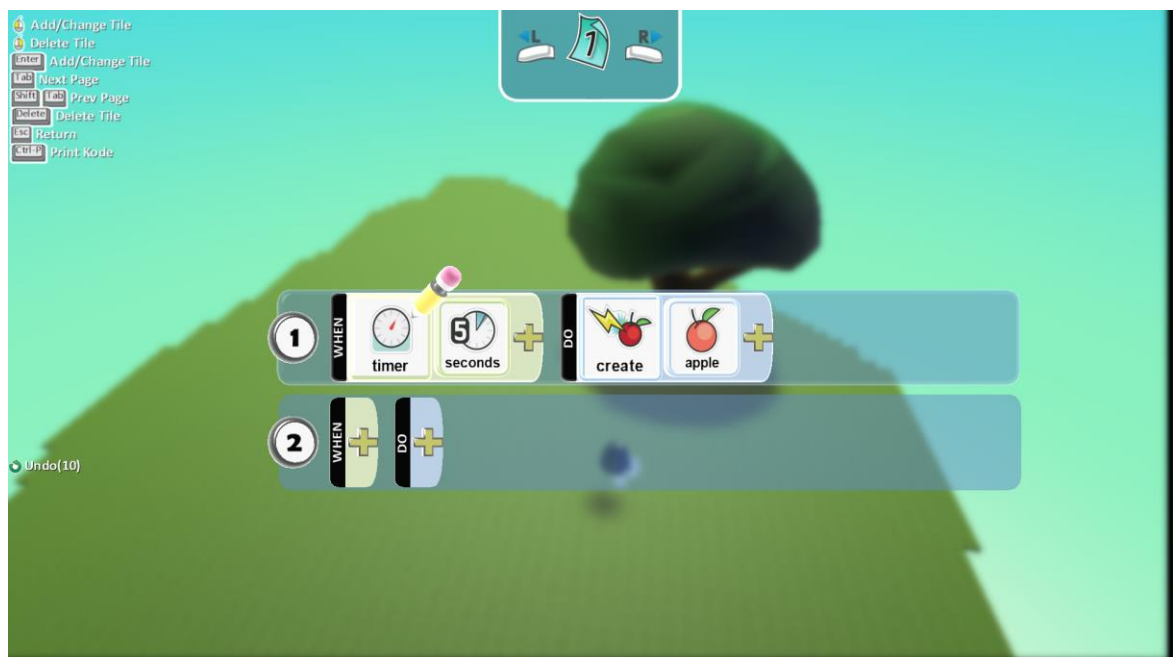


Kuva 21. Ohjelmointi valikko.

Tässä ikkunassa valitaan "When" ja "Do" komentoihin halutut asiat jolloin objekti saadaan tekemään haluttuja asioita (Kuva 21). Nämä kaksi komentoa ovat periaatteessa koko ohjelman tarkoituksen perus. Luodaan syy ja seuraus suhteita ja siten opetellaan perus logiikkaa. Kun sanoja klikataan, saadaan auki valikoita joista pysytään valitsemaan halutut toiminnot. Nyt tehdään puulle seuraavan lainen ohjelma. Eli puu luo joka viides sekunti omenan puun viereen (Kuva 23).



Kuva 22. Eri komentoja ohjelmointivalikossa.

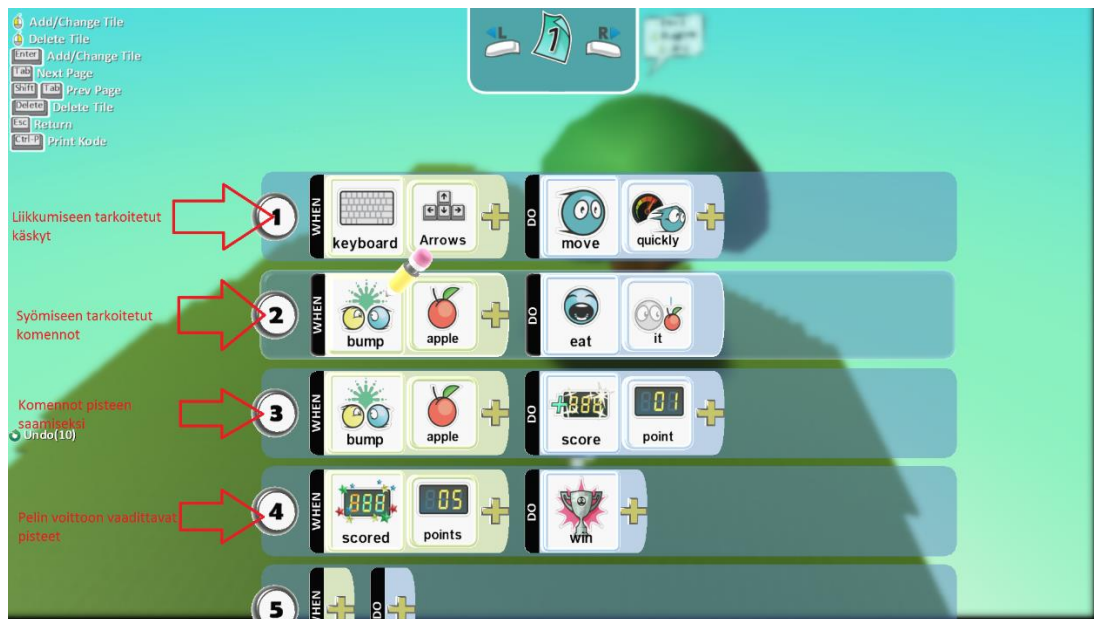


Kuva 23. Puun ohjelma.

Seuraavaksi luodaan ohjelma Kodu -hahmolle joka luotiin. Kodu -hahmon ohjelmointiin kuuluu komentoja joilla saadaan hahmo syömään puun pudottamia omenoita ja syödyistä omenoista pisteitä. Peli päättyy kun hahmo on saanut viisi pistettä.

Ensimmäiseksi on tärkeää että hahmolle ohjelmoidaan kontrollit liikkumiseen. Nyt kun käytössä on hiiri ja näppäimistö, niin valitaan kontrollit sen mukaan. Valitsemalla eri valikoista oikeat vaihtoehdot ja niitä yhdistelemällä saadaan haluttu toiminto.

Kodulle ohjelmoidaan myös käskyt omenan poimimiseen, syömiseen ja pisteen saamiseen seuraavanlaisesti



Kuva 24. Ohjelmoinnin käskyt.

Eli vastaava ohjelmointi selitettynä menee näin (Kuva 24)

1. Kun näppäimistöstä painetaan nuolinäppäimiä, niin silloin Kodu liikkuu nopeasti. (Liikkumisen nopeutta voidaan myös muokata toisen valikon kautta johon palataan myöhemmin.)
2. Kun Kodu törmää omena, se syö sen.
3. Samoin kuten ensimmäisessä eli kun törmätään omena, mutta se antaa yhden pisteen. (Ohjelmointi tapahtuu aina riveittäin, eli moni samankaltainen toiminto tarvitsee tehdä omalle rivilleen.)
4. Kun hahmo on kerännyt viisi pistettä, niin peli päättyy ja hahmo on voittanut pelin.

Tämän session jälkeen käyttäjän pitäisi ymmärtää perus työkalut logiikan ja objektien luonnissa. Sessioita voidaan aina totta kai muuntaa halutun laiseksi. Mutta toisen session idea on yleisesti opettaa ymmärtämään ohjelman työkalut perus logiikan luonnissa.

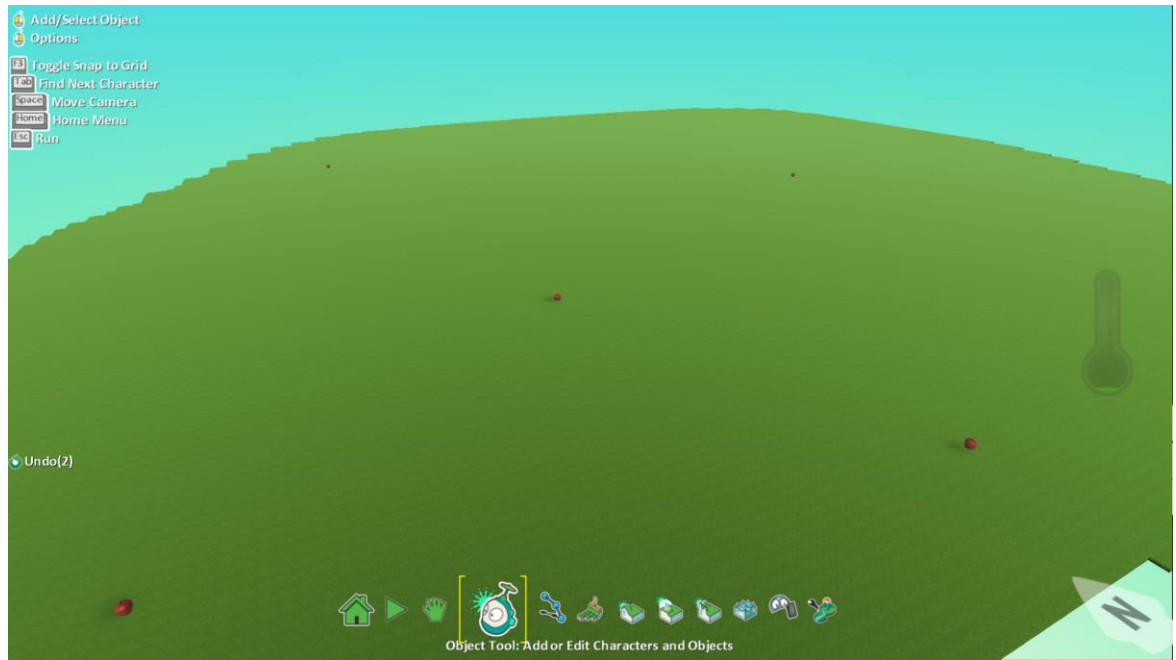
Tilannetta voidaan myös muuttaa siten että omenat ovat jo valmiina maassa eivätkä putoa puusta. Tällöin saadaan helpotusta tekemiseen jos aika ei riitä tai haluaa tehdä asiat yksinkertaisemmin.

2.3 Kolmas sessio. Kaksinpeli

Kolmannella sessiolla käydään läpi maailman ja hahmojen erikoisasetuksia, tutustutaan kaksinpeliin joissa on muita vastustajia.

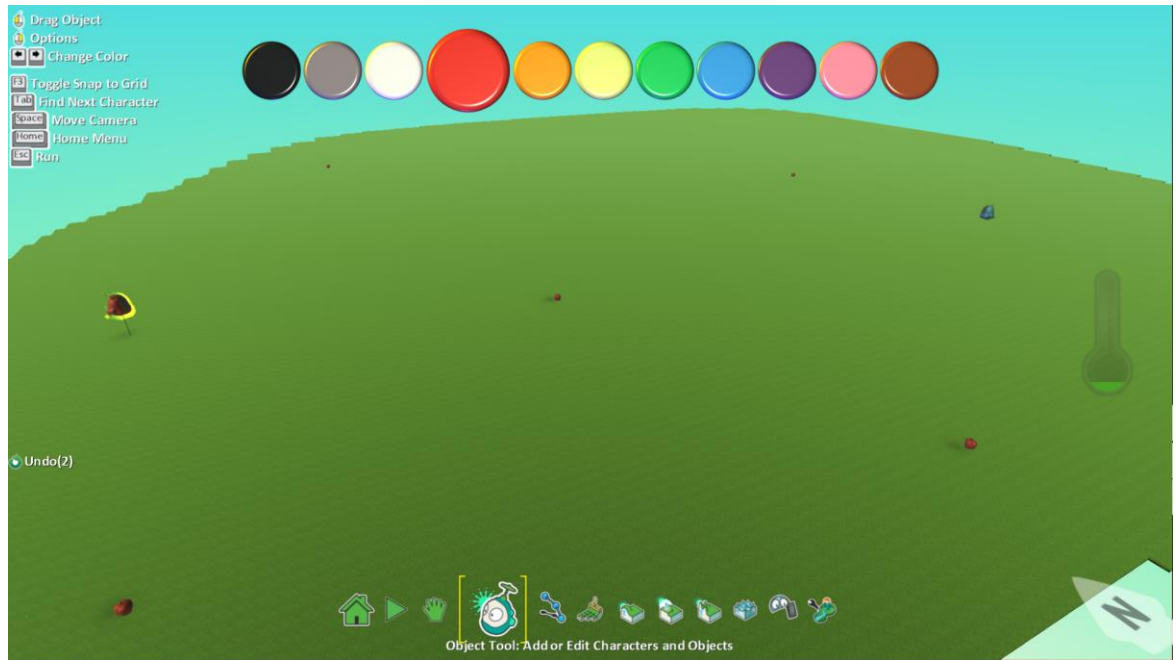
Kodussa on mahdollisuus moninpelille jos käytössä on esimerkiksi Xbox ohjaimia, jolloin pelaajia saadaan samaan maailmaan neljä samaan aikaan. Tällä keralla tehdään peli samalla tavalla kuin toisella kerralla, mutta lisätään maailmaan toinen pelaaja ja yksi tietokoneohjattu vastustaja joka yrittää estää pelaajien etenemisen.

Tälläkin kertaa voidaan aloittaa tekeminen sillä, että valitaan uusi maailma ja laajennetaan että saadaan tilaa. Sen jälkeen voidaan tehdä vaikka viisi omenaa eripuolille karttaa.



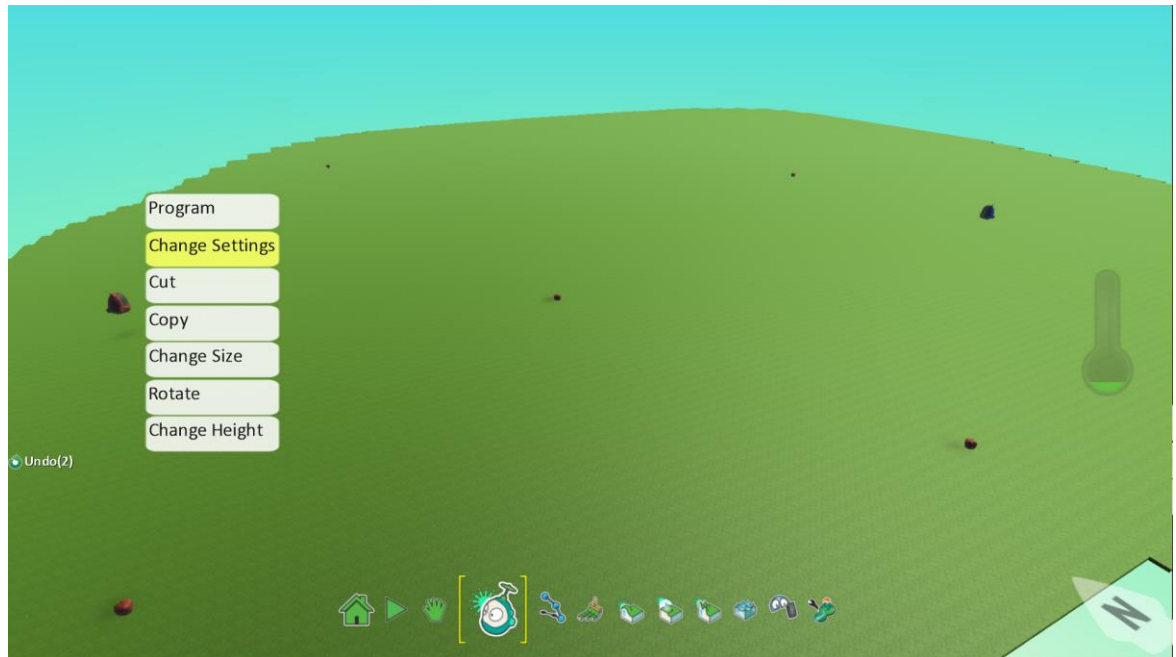
Kuva 25. Viisi omenaa kentällä.

Ensimmäisen vaiheen jälkeen lisätään kentälle kaksi Kodu -hahmoa ja vaihdetaan niiden värit, että tunnistetaan pelaajat väriksen avulla. Hahmojen väritä muutetaan viemällä kursori hahmon päälle ja valitaan nuolinäppäimillä haluttu väri. Tässä tapauksessa on valittu punainen ja sininen (Kuva26). Lisäysten jälkeen tehdään molemmille ensin liikkumiseen tarkoitetut käskyt, mutta muutetaan hahmon nopeutta vieläkin lisää hahmon oman asetusten kautta.



Kuva 26. Kahden pelaajan hahmot erivärisinä.

Hahmon asetuksiin päästään samalla tavalla viemällä kursori hahmon päälle ja valitaan klikkauksen jälkeen valikosta "Change Settings" jolla päästään hahmon valikkoon (Kuva 27). Valikosta voidaan siten muuttaa erilaisia asetuksia kyseiselle hahmolle tai objektille tarvittaessa. Tässä tapauksessa muutamme hahmojen nopeuksia siirtämällä neljän ensimmäisen "säätimen" pykälää kuvan mukaisesti (Kuva 28).



Kuva 27. Hahmon ensimmäinen valikko.

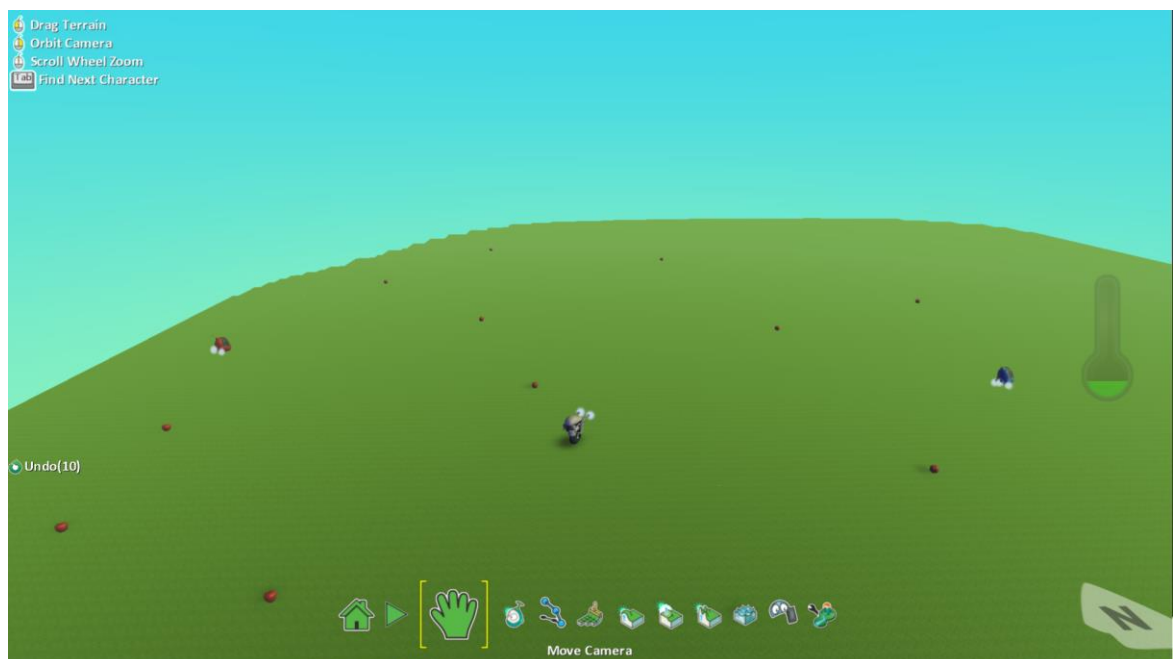


28. Hahmon asetukset ja ”säätimien” asetukset hahmon nopeudelle.

Näiden jälkeen hahmojen pitäisi liikkua vieläkin nopeammin kuin normaalisti. Asetuksia voi muutenkin muuttaa haluamallansa tavalla. Valikoista löytyy kaikenlaisia erilaisia asetuksia, kuten aikaisemmin mainittiin. Jokainen voi näihin tutustua itse ja katsoa mitä tapahtuu.

Asetusten jälkeen tehdään yksi vastustaja joka yrittää jahdata Koduja ja törmätessään jompaankumpaan Kodu -hahmoon, törmätty hahmo häviää pelin ja tuhoutuu. Peli voitetaan vain jos jompikumpi Kodu kerää ensin viisi omenaa. Vastustaja voi olla mikä tahansa liikutettava objekti, tällä kertaa käytetään siihen Cycle-hahmoa (Kuva 29).

Jokainen varmaan huomaa tässä kohdassa ongelman myös että viisi omenaa ei riitä maailmaan jos molempien tarvitsee kerätä viisi omenaa voittaakseen pelin, eli lisätään maailmaan lisää omenoita (Kuva 29).



Kuva 29. Lisää omenoita ja vastustaja joka jahtaa Koduja.

Seuraavaksi käydään läpi objektien ohjelmoinnit ja tarkastetaan että ne ovat oikein. Aloitetaan ensin Kodujen ohjelmoinnin läpi käyminen.



Kuva 30. Punaisen kodun ohjelmointi.

Käydään jälleen läpi jokainen rivi että mitä ne tarkoittavat ja tekevät (Kuva 30).

1. Kun pelaaja 1 käyttää ohjaimen vasenta tattia, niin Kodu liikkuu. Tässä pisteessä täytyy myöskin määrittää, että mikä Kodu liikkuu koska kyseessä on kaksi hahmoa. Tämä on pelaaja 1 jota ohjataan Xbox ohjaimella.
2. Kun jälleen törmätään omenaan, hahmo syö sen.
3. Hahmo saa syömästään omenasta pisteen, mutta tällä kertaa punaisen pisteen että voidaan erottaa hahmojen pisteytys toisistaan.
4. Kun on saatu viisi punaista pistettä, niin pelaaja yksi on voittanut pelin.

Tämä pätee samoin myös toiselle hahmolle, mutta tässä tapauksessa toista hahmoa ohjataan näppäimistöllä koska ohjaimia on vain yksi. Esimerkkinä seuraavassa kuvassa kuinka sama on tehty näppäimistölle (Kuva 31).



Kuva 31. Sinisen kodun ohjelmointi.

Hahmolla on periaatteessa aivan sama ohjelmointi ja lopussa on "Player 2" jolla tarkoitetaan toista pelaajaa tässä vaiheessa. Ero on siinä kun näppäimistön/hiiren kanssa pelataan, niin ei valita pelaajia koska näillä ohjaimilla voi olla aina vain yksi pelaaja. Xbox ohjaimia voidaan määrittää neljä kappaletta. Pelaaja on kuitenkin aina pakko määrittää lopussa jos erotellaan pelaajat voittoa tai muuta toimintoa varten.

Seuraavaksi käydään läpi vastustajan komennot (Kuva 32).



Kuva 32. Vastustajan komennot.

Eli kun hahmo näkee Kodun, niin se lähtee liikkumaan sitä kohti ja törmätessään siihen tuhoaa sen. Tässä tilanteessa ei ole väliä mihin Koduun vastustaja törmää koska maailmassa ei ole muita hahmoja. Yleisesti täytyy määrittää aina ehdot hahmojen tai objektien perusteella että mitä tapahtuu.

Kolmannen session tarkoitus oli opettaa ymmärtämään eri hahmojen vaikutus peliin ja kuinka saadaan eri hahmojen tekemiset vaikuttamaan pelin kulkuun. Neljännellä kerralla tehdään haastavampi peli missä hyödynnetään aikaa ja muita muuttujia. Hyödynnetään myös kaikkea tähän mennessä opittua.

2.4 Neljäs sessio. Haastavampi peli.

Neljännän session aiheena on laajempi ja haastavampi peli. Neljännän kerran tekemiset voidaan myös jakaa useammalle kerralle sen haastavuutensa takia.

Tarkoitus on rakentaa peli jossa puu ns. ampuu omenoita tietyn ajan puitteissa. Pelissä käytetään kahta eriväristä omenaa hyödyntäen pisteytystä. Toisesta omenasta saadaan pisteitä, kun taas toisesta menetetään. Peli päättyy tietyn ajan kuluessa.

Työ voidaan aloittaa sillä, että aloitetaan uusi maailma ja rakennetaan siitä vähän isompi mitä se aloitettaessa on, että puu pystyy ampumaan omenat hallitusti koko alueelle. Tämän jälkeen voidaan luoda yksi puu ja kaksi omenaa, esimerkissä käytän nyt punaista ja vihreää omenaa havainnointiin (Kuva 33).



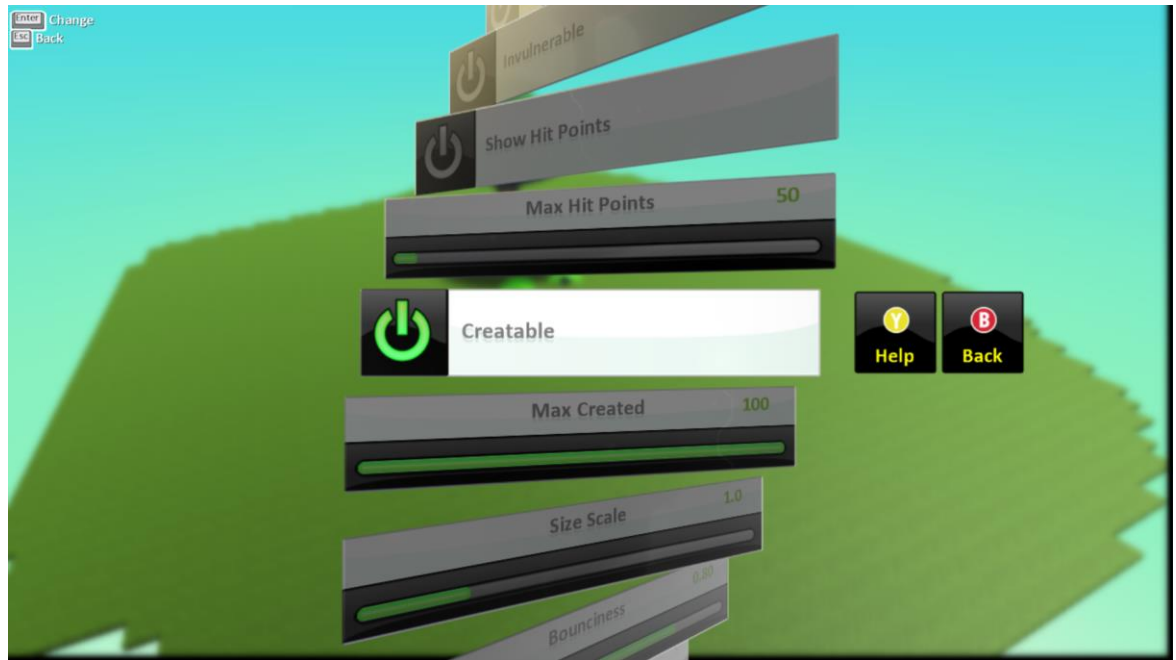
Kuva 33. Esimerkki omenat ja puu.

Kuvassa omenat hohtavat koska niiden asetuksia on muokattu. Asetuksia on muokattu siten että objekti voi luoda itsensä uudelleen ja uudelleen. Tämä on hyvä asetus silloin kuin halutaan tehdä peli, missä esimerkiksi jokin hahmo tai objekti täytyy luoda uudelleen tietyn väliajoin. Asetus vaihdettiin objektin omasta valikosta menemällä sinne ensin objektia valitsemalla ja klikkaamalla oikeaa hiiren painiketta jolloin päästään valitsemaan objektin oma valikko (Kuva 34).



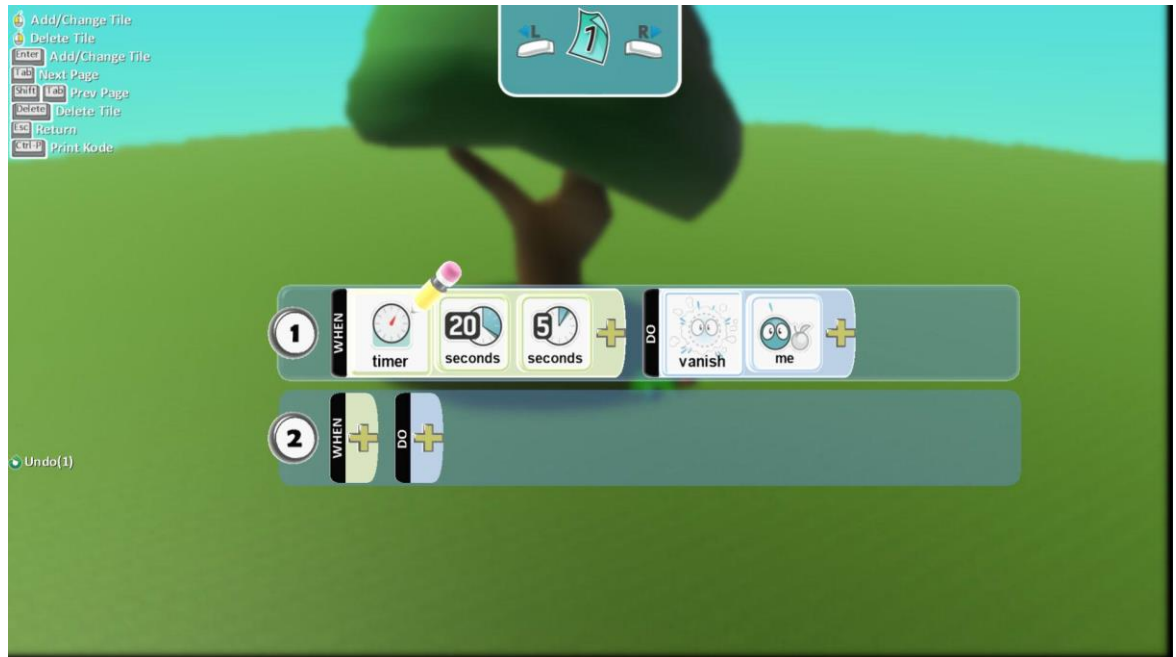
Kuva 34. Objektin omaan välikoon meneminen.

Omasta valikosta sitten etsitään kohta "Creatable" ja klikataan päälle. Tämän jälkeen sen alle ilmestyy vaihtoehto jolla voidaan määrittää luotavien objektin määrä. Annetaan se olla vain tässä kohtaa 100, tämä vain osoittaa sen että kuinka monta objektia voidaan luoda maailmaan samaan aikaan (Kuva 35). Automaattisesti luotavien objektien huono puoli on se, että ne nopeasti kuluttavat maailman mittarin täyteen jolloin ei voida enää luoda mitään tai peli hidastuu, jos kyseessä on hitaampi tietokone.



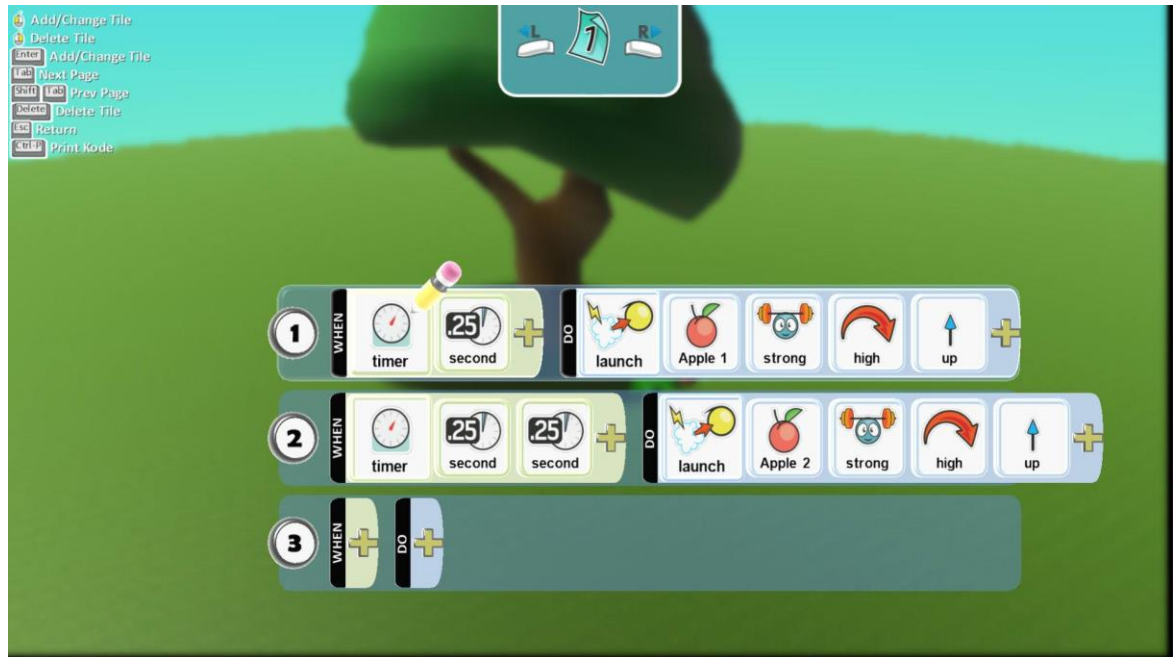
Kuva 35. Asetukset.

Tämän jälkeen mennään takaisin ja valitaan toinen omenoista, mutta tehdään kuitenkin molemmille omenoille sama ohjelmointi. Koodi on hyvin yksinkertainen itse omenoille. Siinä määritetään että kun tietty aika on kulunut, niin omena häviää. Tämä siitä syystä että maailma ei tulisi liian täyteen ja samalla estetään pelin muodostuminen liian raskaaksi itse tietokoneelle ja järjestelmälle (Kuva 36).



Kuva 36. 25 sekunnin ajastus omenan häviämiseksi (molemmille omenille).

Tämän jälkeen valitaan puu ja aletaan tekemään sille ohjelmointia. Puu ohjelmoidaan ampumaan omenoita tietyn ajan välein. Mutta puuta ohjelmoidessa käytetäänkin tällä kertaa normaalin omenan sijasta omenoita jotka ohjelmoitiin häviäväksi tietyn ajan puitteissa. Tämä on pakko tehdä, koska ohjelma ei itse osaa häivyttää omenia ja muuten jokaiselle luotavalle omenalle olisi tehtävä sama ohjelmointi, mikä on käytännössä mahdotonta. Näin ns. merkitsimme omenat "Creatable" komennolla, niin ohjelma ymmärtää että kyseessä on tietyt omenat joita puu myöskin ampuu.



Kuva 37. Puun koodi.

Eli ohjelmointi puulle tapahtuu seuraavanlaisesti (Kuva 37)

1. Ajastin ampuu 0,25 sekunnin välein omena numero yhden. Tässä on määritelty myös että omena ammutaan korkealle ja ylös. Idea on että kun paljon omenoita ammutaan pienen ajan sisällä, niin ne saadaan putoamaan eripuolille kenttää.
2. Tässä ammutaan omena numero kaksi samalla tavalla kuin numero yksi, mutta ajastimeen on lisätty toinen 0,25 sekuntia jolloin toinen omena ammutaan 0,50 sekunnin välein.

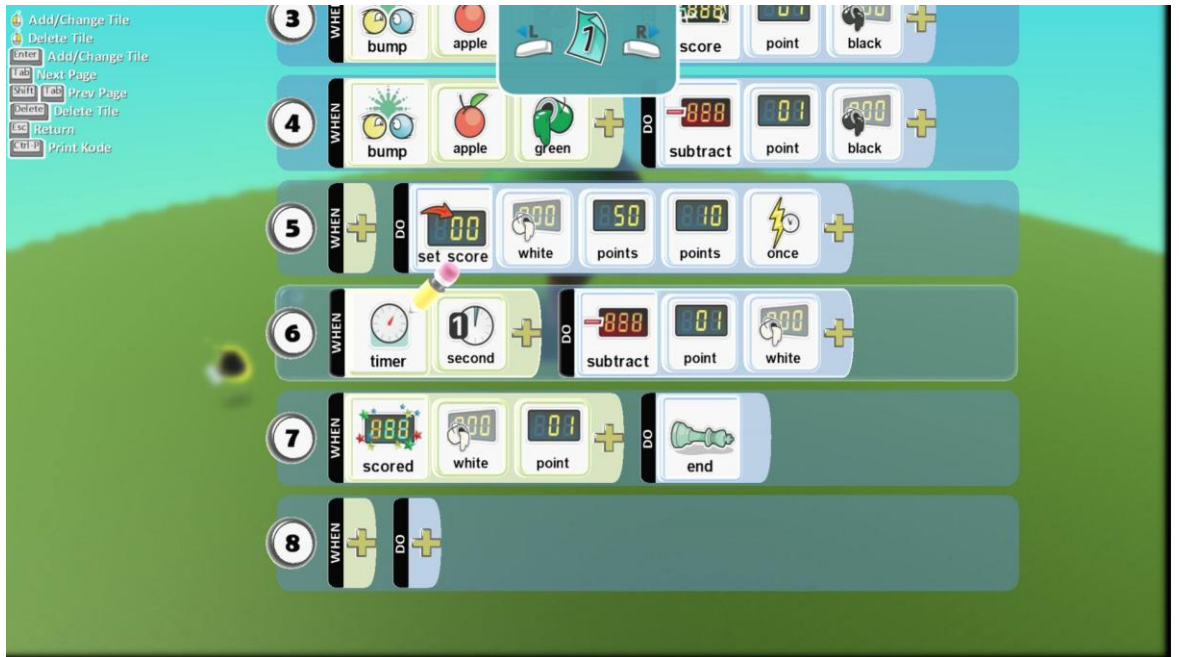
Omenat on merkitty 1 ja 2, valinta tapahtuu kun valitaan ohjelmointiin objektin siasta "Creatables" josta löytyy Apple 1 ja Apple 2. Nämä ovat ne erikoisomenat jotka luotiin aikaisemmin. Jos valitaan objekti kohdasta omena, niin ohjelmointi ja peli eivät tule toimimaan siten kun halutaan.

Tämän jälkeen lisätään kenttään yksi Kodu -hahmo. Isoin ohjelmointi tulee tapahtumaan sille seuraavanlaisesti Xbox -ohjainta käyttäen (Kuva 38 & 39).



Kuva 38. Kodun ohjelmointi osa 1.

1. Vasenta tattia liikuttaessa hahmo liikkuu.
2. Kun törmätään vain johonkin omena, hahmo syö sen.
3. Kun törmätään punaiseen omena, pelaaja saa siitä yhden mustan pisteen.
4. Kun törmätään vihreään omena, se vähentää yhden mustan pisteen



Kuva 39. Kodun ohjelmointi osa 2.

5. Tässä asetetaan pisteet ikään kuin ajaksi. Eli heti kun peli alkaa, niin maailmaan tulee ns. 50 pistettä ja 10 pistettä joista muodostuu 60 pistettä/sekuntia.
6. Joka sekunti miinustetaan yksi valkoinen piste joka kuvaa aikaa ja tässä tapauksessa sekuntia.
7. Kun valkoisia pisteitä on enää yksi jäljellä, peli päättyy jolla kuvataan ajan loppumista.

Ohjelmassa ei itsessään ole suoraa ajastinta pelin päättymiselle ja se tarvitsee-kin tehdä aina tällä tavoin.

Tästä pelistä voidaan tehdä myös kaksinpeli jolloin voidaan kilpailla kumpi kerää enemmän omenoita.

Eli Neljännen session tarkoitus on saada käyttäjä ymmärtämään ajan ja monimutkaisempien muuttujien ominaisuudet. Kun sessioiden ajaksi on määritelty puolitoistatuntia, niin voi neljännen session sisäistämiseen mennä joissakin tapauksissa myös pidempi aika jolloin voidaan venyttää sessiota seuraaville kerroille.

3 MITÄ TEHDÄÄN ALKEIDEN JÄLKEEN?

Oppaassa kuvattujen alkeiden jälkeen käyttäjä ymmärtää ja osaa ohjelman peruslogiikan ja siihen tarvittavat työkalut sen toteuttamiseen. Ymmärrystä on myös erilaisten muuttujien vaikutuksista itse peliin. Näitä hyödyntämällä kannattaa lähteä tutustumaan vielä muihin työkaluihin lähemmin koska nyt tiedetään mistä ne löytyvät ja mitä niiden kanssa voidaan suurin piirtein tehdä.

Lisää vinkkejä tekemiseen ja toteutukseen löytyykin juuri Kodun omalta sivustolta osoitteesta <http://www.kodugamelab.com>. Sivusto on hyvä syystä, että sieltä löytyy vielä lisää harjoituksia opetteluun ja muuhun kokeiluun, sivusto on englanninkielinen. Sivuilta löytyy myös hyvä foorumi ja kommuuni tekijöille. Foorumilta löytyy infoa yleisimpiin ongelmatilanteisiin muiden käyttäjien avustamana. Näiden kaikkien lisäksi sivuilta pystyy vielä lataamaan muiden käyttäjien tekemiä pelejä ja harjoituksia.

Myös YouTubesta (<http://www.youtube.com>) löytyy hyviä ja havainnollistavia ohjeita kokeiluun videoiden kera.